8. Problem Management Procedures

ECS Problem Management is administered through system-level and site-level control board reviews. These control boards oversee the analysis, recommendations, and actions taken to resolve ECS system/site problems concerning hardware, software, documentation, and procedures. The M&O and the site-level maintenance organization resolve routine maintenance issues at the system-level and site-level, respectively, using the Trouble Ticket System for tracking maintenance changes. Trouble Tickets (TT) may evolve into Non Conformance Reports (NCR), as required, which may then be utilized to generate Configuration Change Requests (CCR) to effect changes to the approved baseline. To ensure controlled change, NCRs are tracked using the DDTS in the EDF and CCRs manually by Systems Engineering.

The Trouble Ticket System is the first vehicle used to record and report problems with the operational system. Trouble Tickets can be generated by operations, maintenance, development, and customer personnel as well as users. The Trouble Ticket System is an automated database that tracks the resolution activities associated with each trouble ticket. Documentation that is related to the problem, and is not in electronic form, or is in electronic form at the DAAC, is distributed by the local CM Administrator (CMA), and is listed as an attachment to the trouble ticket.

The CM Administrator at each site serves as Trouble Ticket System administrator. Trouble Tickets initially generated at a site, the resolution of which require changes to the system level baseline, are forwarded to the SMC, where they are reviewed and *translated* into NCRs. Additionally, Trouble Tickets and CCRs that are generated at the sites, which are repaired locally, and result in site-unique extensions to the system level baseline, are forwarded to the SMC for tracking across the ECS baseline. The SMC CMA is responsible for tracking ECS level TTs after they have been received from the sites, and for propagating system problem resolutions for site visibility.

CMAs also support the activities of the local Problem Review Board. This includes generating status reports, and implementing resolutions, instructions, and changes as directed by the Board. User Services Representatives monitor trouble tickets to notify users concerning problem resolution and status. Maintenance engineers at respective levels will record all activities in the trouble ticket. This information can be used to determine critical maintenance concerns related to frequency of occurrence, criticality level, and the volume of problems experienced. The maintainability analysis will guide critical changes, volume and type of support components to be utilized, and will focus further ECS release development.

This section provides an overview of the Trouble Ticketing process and defines the M&O procedures for processing and resolving trouble ticket submissions. In addition, this section provides instructions for diagnosing network problems.

8.1 The Problem Resolution Process — An Overview

8.1.1 ECS Internal Process

Any ECS user may submit Trouble Tickets using the local Trouble Ticketing System (TTS) at any site. TT submission triggers an internal review by the site's review board. Primary objectives of the internal review are to quickly identify and correct problems that fall within the site's capability to maintain, review and validate the priority of the problem, and to elevate to the system level those problems that either exceed their capability to repair, or that require a change to the system level baseline.

Problems passed from the sites to the ECS system level, are passed by transferring the trouble ticket from the local TTS to the SMC TTS. Here they are reviewed by the M & O Problem Review Board (PRB), which hosts daily teleconferences, known as the PRB Telecon.

The permanent membership of the Problem Review Board is as follows:

a. Chair: Problem Management Lead or designee

b. Each DAAC: one member representative

c. Quality Assurance: one member representative

d. ECS Integrated Logistics Support: one member representative

e. Secretary: SMC CM Administrator

f. Systems Engineering Department: one member representative

g. ESDIS M&O: one member representative

The roles and responsibilities of the participants in the PRB Telecon are to:

a) Follow a nominal agenda that includes the following discussions:

i) new NCRs

ii) deferred trouble tickets

iii) Review NCRs in the Verified State of DDTS.

b) Review severity of each NCR according to the guidelines specified in the

Operations Class NCR Management Process (MO-1-003-5)

c) Convert any TTs that identify a system non-conformance and have the appropriate information will be forwarded into a NCR by requesting that the CM Administrator forward the TT to the DDTS (Distributed Defect Tracking System). NCRs document the system non-conformance and are worked by the Sustaining Engineering staff.

d) NCRs that are deemed to not have enough information for proper adjudication will either be rejected as incomplete or they will be deferred until a future meeting as determined by the

PRB chairperson. The PRB administrator will maintain a list of deferred items and place those items on a future PRB agenda after more information is gathered.

The PRB performs a preliminary review of each trouble ticket to confirm the priority assigned by the site, the completeness of information and data relevant to the problem, and whether it requires a change to the system level operational baseline.

The PRB may forward TTs to the appropriate organizations or individuals for further analysis and technical investigation, resolution proposal, and NCR preparation, when required. The CM Administrator or Daac Support Help Desk members provide administrative support to the PRB by publishing minutes of teleconference meetings, tracking TTs and action items, and by translating TTs to NCRs.

The PRB has the authority to direct resolutions to trouble ticket problems that do not change, or in any way affect, the ECS operational baseline and baseline documentation. An NCR is required when the Technical Investigation (TI) determines that the operational baseline must be changed in order to correct the problem identified in the trouble ticket.

The PRB is not a voting board; the membership is appointed for the purpose of providing timely, direct technical support to the Chair, who has the decision making responsibility and authority.

The M&O CCB has the authority and responsibility to approve Class II changes to the operational baseline. Specific responsibilities include:

a. Review, approve and schedule; review and backlog; or reject each NCR's proposed resolution, or cost and schedule input from the Responsible Engineer (RE).

b. Approve the schedule for the deployment of configuration changes in the form of a 'drop' to the SMC.

c. Approve the content of each block.

d. Manage and adjust the schedule and contents of each block in accordance with ECS SDPS Program priorities and the progress of NCR work-off.

e. Review the status of all backlogged NCRs on a periodic basis. Schedule NCRs for a future block as appropriate

f. Collect and report on NCR statistics.

The permanent membership of the M&O CCB is as follows:

a. Chair: ECS M&O Manager of designee.

b. Each DAAC: one representative.

c. SMC: one representative.

d. System Engineering Department: one representative

e. Development Department: one representative.

f. The Patch IPT (or Test Department): one representative.

g. ECS Integrated Logistics Support Department: one representative.

h. Secretary: TBD

The M&O CCB reviews each NCR as received from the PRB for technical merit, completeness, priority and complexity. It may direct additional technical investigation and analysis, or assign immediately for resolution. The Responsible Engineer prepares a proposal which will include:

a. A short, narrative description of the change that identifies the object or objects, configuration files, COTS SW, documentation, or scripts that require change. The nature of the change must also be included in the narrative.

b. A cost estimate where:

    1. LOW requires fewer than 5 staff-days of effort

    2. MEDIUM requires 6 to 20 Staff-days of effort

    3. HIGH requires more than 20 staff-days of effort

c. The date by which the change could be ready for integration and deployment, given the RE's knowledge of tasking, priorities, development activities, etc.

The M&O CCB is not a voting board. The CCB Chair has the singular authority for decision making. Other board members are appointed to provide direct and timely technical support to the CCB's decision making process.

In accordance with ECS Configuration Management procedures, the RE, as directed by the CCB, makes the necessary changes to the system baseline, including the ECS baseline documentation changes to support the resolution. These changes are then integrated into an M&O block by the Patch IPT.

The Patch IPT integrates and tests each block to verify that the system still meets functional and performance requirements, that the integrated block corrects the NCR(s), and that new features execute properly. The status of each block passing through integration and test is reported to the CCB regularly by the Patch IPT.

Upon satisfactory completion of the test program, the M&O CCB may authorize deployment of the block to the DAACs. The SMC pulls the block from the EDF and saves it to a locked-down directory. The SMC then notifies the DAACs that the block is available for transfer. Each DAAC desiring the block notifies the SMC of the transfer path, and the SMC pushes the block as requested. DAACs may then install the block and keep the SMC apprised of the state of each mode.

8.1.2 Interface with ESDIS

The EOSDIS Sciences Systems Configuration Management Board CCB must approve all ECS Class I CCRs before work is authorized to commence. (See the *Sciences Systems Configuration*

Management Board (PCMB) Configuration Management Plan (level 3) for discussion of change classification.)

Should the RE recommend, and the M&O CCB concur, that a CCR is an enhancement rather than a maintenance change, the M&O CCB will place the CCR in backlog after reviewing and approving the technical solution, cost and schedule impact. The CCR is then sent to the PCMB for approval. When approved by the PCMB and returned to the M&O CCB, the CCR is assigned to an RE for implementation.

When the PCMB or the ESDIS CCB forwards a new CCR as an enhancement to the M&O CCB, the CCR is assigned to an RE by the CCB. The CCB approved technical, cost and schedule assessments of the RE are sent by the M&O CCB back to the PCMB.

8.2 Problem Management Procedures

The Trouble Ticket System is comprised of the Remedy Action Request System, a Commercial Off-The-Shelf (COTS) product that provides a distributed trouble ticketing service which provides a common environment and means of classifying, tracking, and reporting problem occurrence and resolution to both ECS users and operations personnel. The trouble ticketing service:

- Provides a GUI for operations personnel to access all Trouble Ticket services.
- Provides a common Trouble Ticket entry format.
- Stores Trouble Tickets.
- Retrieves Trouble Tickets via ad hoc queries.
- Allows operations personnel to forward problems from one DAAC to another.
- Produces stock and common reports.
- Provides an interface to user’s and operator’s e-mail to provide automatic notification.
- Offers an application programming interface through which applications can submit trouble tickets.
- Provides summary information to the SMC from each DAAC to allow trend reports regarding trouble tickets.
- Defines a consistent “life cycle” for trouble tickets.
- Allows each DAAC a degree of customization through definition of further re-prioritization and action rules.

In addition to the functionality provided by Remedy’s Action Request System, the Trouble Ticketing Service utilizes a set of custom HTML pages ("screens") to provide registered users with the ability to submit new trouble tickets and query the current status of any of their previous entries. Access to the Trouble Ticketing System through this technique provides users an easy method for reporting problems in an environment with which most are already familiar.

Additionally, as another means of trouble ticket entry, the Trouble Ticket System provides a text e-mail template through which automated entry of trouble tickets is possible. Support staff members are able to enter Trouble Tickets through the TTS interface for problems received via other methods (for example, phone calls).

The Remedy Action Request System also functions as the User Contact Log. Remedy's Action Request System is configured to have a separate form that contains the entries that User Services personnel enter for each contact that they receive from a user. The User Contact Log allows a trouble ticket to be initiated from a log entry: with the push of a button — the Trouble Ticket will be populated with information from the contact log.

Users submit trouble tickets to the User Services Desk. External users submit trouble tickets through the Internet [using a series of hypertext mark-up language (HTML) screens]. Site personnel submit trouble tickets via the Trouble Ticket System (Remedy).

User Services personnel process trouble tickets through the Trouble Ticket System for problem resolution in accordance with local policy. Trouble tickets are first evaluated to determine the severity of the problem and assignment of on-site responsibility. Every trouble ticket is logged into the database for record keeping purposes. Trouble tickets that can be resolved locally are assigned and tracked at the local center. The Operations Supervisor reviews each trouble ticket for priority verification and problem description; and assigns it to an appropriate Maintenance Engineer for resolution.

Matters that require external or higher level assistance; and problems, the repair of which require changes to the system baseline; are escalated to M&O via the Problem Review Board (PRB) Telecon for discussion and disposition. The telecon is held to coordinate trouble ticket activities within the M&O organization as well as with development, customer, and user organizations.

The CM Administrator or Daac Support Help Desk staff is responsible for preparing and disseminating the agenda and the minutes for the PRB . Electronic dissemination via the web or e-mail is preferred. Agenda items may be supplemented or replaced by hardcopy or softcopy reports. Material from this meeting is distributed within each ECS organization and to customer and user organizations as required. A typical agenda might include:

- Review and prioritize each trouble ticket opened at each center.
- Review and re-prioritize older trouble tickets (as required).
- Assign trouble ticket work-off responsibility to one organization.
- Review distribution of trouble tickets by organization, priority and age.
- Discuss trouble ticket issues with development organizations.

8.3 Using the Trouble Ticket System

1. User or Operator discovers a problem with ECS (hardware, software, documentation, procedure) and documents this problem for later resolution. The submitter forwards a trouble ticket to User Services by: calling up the Trouble Ticket System via the Internet;

going on-line with the Trouble Ticket System; phoning User Services; or sending an e-mail message to the Trouble Ticket System.

2. The trouble ticket is logged into the system. TTS automatically assigns "New" status to the trouble ticket and notifies the Operations Supervisor for assignment and prioritization. TTS notifies the Operations Supervisor via email, or through Remedy's notification tool, or both. The status of each trouble ticket, as it progresses through the resolution process, is recorded by the CMA.

3. Categorizing Problem Severity. The Help Desk Primary or Secondary will review the information provided by the DAAC to determine if the problem has been described in enough detail to warrant the recommended severity. If there is insufficient information, the Help Desk Primary or Secondary will contact the DAAC submitter or the DAAC representative to collect the necessary details to categorize the problem. In the case where a problem was submitted as a Medium or Low Impact Trouble Ticket and the Help Desk has determined that there is insufficient detail to provide a proper assessment, the Trouble Ticket will be deferred to the Problem Review Board for further review.

   In determining severity of the problem, the Help Desk must consider the following factors:

   - Impact on the ability to ingest, process or distribute satellite data
   - Frequency of occurrence
   - Availability of and adequate work-around

Refer to the following figure (Figure 8.3-1) to determine problem severity:

As Documented in NASA 420-05-03	As Used/Interpreted by M&O
Category 1: System/Service cannot perform critical function or imposes major safety hazard. (Priority 1) Presents an immediate impact to development, operations, services, or data processing functions; imposes major safety hazard to personnel, systems, or space mission resources; or results in loss of one or more essential mission objectives.	**HIGH (Severity 1)**: An NCR which causes: – Inability to perform a mission-critical function (i.e., Ingest/Pre-Processing/Archiving of Science Data, Planned Processing, Browse/Order/Distribute); – Performance of a mission-critical function to be so degraded that production minimum goals cannot be achieved; – A mission-critical function to be performed improperly, resulting in permanent loss of data; ***and*** for which no workaround exists ***or*** for which no workaround can be accommodated by DAAC operators given a detailed workaround procedure is documented but the procedure is inadequate based upon the complexity of the procedure, the abilities of an adequately trained and experienced operator, or both.
Category 2: System/Service substantially impaired. (Priority 2) Substantially impacts development, operations, services, or data processing functions; fails to operate within critical performance specifications; or cannot effectively or efficiently fulfill baseline requirements.	**MEDIUM (Severity 2)**: An NCR with the consequence that: – The performance of a mission-critical function is degraded and may prevent achieving production minimum goals; – A mission-critical function can be only partially performed, or performs improperly, resulting in temporary loss of data or incorrect data results; – A situation (actually or potentially) severly compromises ECS mission readiness or operational integrity; – A condition exists to produce a severely degraded mission-critical function, but a workaround will allow operations to continue temporarily without permanent loss of data or severely impaired

Figure 8.3-1. Trouble Ticket Priority/NCR Severity

***Mission-Critical functions:**

Ingest/Pre-Process/Archive Science Data
Planned Processing
Browse/Order/Distribute

**** Non-Mission-Critical functions**

The Trouble Ticket System (TTS) has coded these three priorities as HIGH, MEDIUM, and LOW. All trouble ticket submittals are required to designate a priority level for the problem. However, the formal priority is assigned by the Operations Supervisor and maintained by the CM Administrator. M&O applies these additional priorities:

Priority 4: Nuisance Problem:

Problems that do not impair the capability of the ECS, but rather could simplify the it's use, such as the arrangement of video screens, color, and so on.

Priority 5: Enhancement:

Problems, the resolution of which result in enhanced system capability, or are out-of scope (Class I problem).

4. All affected Operations Supervisors at the sites (SMC, DAACs, EOC, EDF) are notified by e-mail of the problem and solicited for inputs to problem assessment (impact) and resolution.
5. The Trouble Ticket database is updated by the CM Administrator or the DAAC Support Help Desk Staff. The trouble ticket may be modified to reflect any new information/coordination activity.
6. The Operations Supervisor assigns the problem to a Problem Investigator for further follow-up.
7. The Problem Investigator coordinates input from SEO, developers, vendors, and external organizations to effect the local resolution. The Problem Investigator presents significant issues at the PRB Telecon.
8. The Problem Investigator updates the trouble ticket database.
9. The Problem Investigator forwards any information regarding proposed/implemented fixes to the established notification list.
10. In those cases where the problem resolution will result in a change to the operational baseline, or where the local site wishes to elevate the TT to M&O for advice or for resolution, the site CMA forwards the TT to the SMC.
11. The proposed resolution is then presented to the Problem Review Board for review, ratification, or revision.
12. Changes that do not affect controlled configuration items may be approved and implemented by the Failure Review Board/Problem Review Board and closed.
13. All system level changes are proposed in Non Conformance Reports (NCRs). The SMC is the custodian of a translation utility which, when invoked, opens an NCR in the Change Request Manager, and closes the trouble ticket in TTS. Emergency fixes

(Priority 1) can be made locally with the approval of the local CCB, and then reported to M&O after the crisis is resolved. The M&O CCB may approve, reject, defer or revise the NCR.

14. The off-site problem resolution process is monitored by the M&O Problem Review Board, which may also revise the proposed solution because of any system-level effect(s).

15. The NCR may be escalated to higher level CCBs for system and/or external elements that may be involved in the resolution process.

Table 8.3-1. Trouble Ticket System - Activity Checklist

Order	Role	Task	Section
1	ECS users	Access the Trouble Ticket System	8.3.1, 8.4.1
2	ECS users	Submit Trouble Ticket	8.3.2
3	Maintenance Engineer	Modify Open Trouble Ticket	8.3.3
4	Operations Supervisor, Maintenance Engineer	Forward Trouble Ticket	8.3.4
5	Database Administrator	Add Users to TTS	8.3.5
6	CMA	Modify TTS User Privileges	8.3.6
7	CMA	Modify TTS' Configuration	8.3.7
8	Maintenance Engineer, CMA	Generate Reports	8.3.8
9	Maintenance Engineer	Maintain Escalation Time Table	8.3.9

8.3.1 Accessing the Trouble Ticket System

The Trouble Ticket System may be accessed through either HTML or Remedy. The Trouble Ticket HTML is used by both User Services and the end user to submit trouble tickets without going through Remedy. It is accessed through the web. Through HTML, the user can submit, obtain a list, and view details of trouble tickets. Complete and detailed instructions for Remedy may be found in the current DID 609-CD and *Remedy's Action Request System Users's guide.*

Through Remedy, the User clicks on the User Tool icon, which opens the RelB-Trouble Tickets form to submit, query, or work a Trouble Ticket. The Main Remedy Trouble Ticket screen is used to select the appropriate forms for submitting, modifying, or displaying a trouble ticket. The Main Page data fields are identified in Table 8.3.1-1.

The Remedy Action Request System provides a distributed Trouble Ticketing Service that furnishes DAACs a common environment and the means of classifying, tracking, and reporting problem occurrences and resolutions to both ECS users and operations personnel. The Trouble Ticketing Service:

· provides a GUI for operations personnel to access all Trouble Ticket services

· provide a common Trouble Ticket entry format

· stores Trouble Tickets

· retrieves Trouble Tickets via ad-hoc queries

· allows operations personnel to forward problems from one DAAC to another

· generates reports and statistics

· interfaces with user's and operator's e-mail to provide automatic notification

· offers an application programming interface through which applications can submit Trouble Tickets

· provides summary information to the SMC from each DAAC to allow trend reports regarding Trouble Tickets

· enables operations personnel to forward a copy of a "closed" trouble ticket to the SMC for insertion into the ECS Closed Trouble Ticket Database

· defines a consistent "life-cycle" for Trouble Tickets

· allows each DAAC a degree of customization through definition of further escalation and action rules.

Several timesaving features are available through Remedy: the Admin Tool, Remedy Import tool, the Hardware Information form, and the GUI Notification tool. Brief descriptions are provided in Sections 8.3.1.1 through 8.3.1.4.

Table 8.3.1-1. RelB-Trouble Ticket Field Description (1 of 2)

Field Name	Data Type	Size	Entry	Description
Ticket-Id	Character	15	System generated	Ticket number which is set and maintained by the system
Ticket Status	Selection	4	Required	Status of the Trouble Ticket
Assigned-Priority	Selection	4	Required	Priority of Trouble Ticket assigned at the site (HIGH, MEDIUM, LOW)
Short Description	Character	128	Required	Short Description of the problem
Submitter Impact	Selection	4	Optional	Impact of the problem to the submitter (HIGH, MEDIUM, LOW)
Long-Description	Character	4060	Optional	Long Description of the problem
Resolution Log (End User Sees)	Diary	Unlim	Optional	General steps in the resolution of the problem
Detailed Resolution Log	Diary	Unlim	Optional	Detailed steps in the resolution of the problem
Submitter ID	Character	30	Required	User Id of the Submitter.
Submitter Name	Character	30	Optional	Full Name of the Submitter
Submitter Phone	Character	30	Optional	Phone number of the Submitter
Submitter e-mail	Character	64	Optional	E-mail address of the Submitter
Submitter Home DAAC	Character	60	Optional	Home DAAC of the Submitter
History	Diary	Unlim	Optional	Upon submission or modification, the person assigned to the ticket and the ticket status will be indicated in the History field. Due to a limitation in Remedy, this information will only be written when the Assigned-to and Status fields are modified
CI	Character	30	Required	Name of the Configuration Item to which the problem is associated
Assigned-To	Character	30	Optional	Person that Trouble Ticket has been assigned to
Last-modified-by	Character	30	System generated	Person that last modified the Trouble Ticket
Create-date	Date/Time	4	System generated	Date Trouble Ticket was created at the present site
Last-Modified-date	Date/Time	4	System generated	Date the Trouble Ticket was last modified
Related CCR	Character	60	Optional	ID of a related CCR

Table 8.3.1-1. RelB-Trouble Ticket Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
Key Words	Character	255	Optional	Key words to help identify this Trouble Ticket
Problem Type	Character	30	Required	Type of problem addressed by this Trouble Ticket
Closing Code	Character	60	Required	Disposition of the closed trouble ticket
Closed-by	Character	60	System generated	Person that closed this Trouble Ticket
Close-date	Date/Time	4	System generated	Date this Trouble Ticket was closed
Software Resource	Character	60	Optional	Software Resource that the problem came from
Hardware Resource	Character	60	Optional	Hardware Resource that this problem came from
Duplicate Master Id	Character	25	Optional	The Master Ticket-ID of this Trouble Ticket
Forward-to	Character	60	Optional	Site that this Trouble Ticket was last forwarded to
Forwarded-from	Character	60	Optional	Site that forwarded this Trouble Ticket
Forwarded-by	Character	60	Optional	Contact person at the forwarding site
Forward-date	Date/Time	4	Optional	Date Trouble Ticket was forwarded
Unique-Identifier	Character	20	Optional	Unique identifier which is established at the origination site This identifier should NEVER be changed once set
Forwarded-to-1	Character	60	Optional	First site to have been forwarded this Trouble Ticket
Forwarded-to-2	Character	60	Optional	Second site to have been forwarded this Trouble Ticket
Forwarded-to-3	Character	60	Optional	Third site to have been forwarded this Trouble Ticket
Forwarded-to-4	Character	60	Optional	Fourth site to have been forwarded this Trouble Ticket
Associated Contact Log Id	Character	30	Optional	ID number of the Associated Contact Log

8.3.1.1 Remedy's GUI Admin Tool

Most Remedy administrative functions are accomplished using the Remedy Administration tool, implemented in the Windows environment on a personal computer. The log-in dialog provides an **Accounts . . .** button for selecting the user account and servers to be addressed in the administration session. When the account and servers have been selected and the log in is complete, the Remedy Administrator Server Window is displayed. In this window, the server directory structure may be expanded and an object type may be selected for performing administrative functions on various “objects” such as forms and groups (see paragraphs 8.3.5, 8.3.6, and 8.3.7).

8.3.1.2 Remedy Import Tool

The Remedy Import tool is used to import existing entries rather than retyping information manually. It also enables the user to import entries into a form from a file generated by the Admin tool. For more information on the Import tool, refer to the *Action Request System 4.5 Concepts Guide* and the *Action Request System 4.5 Workflow Administrator’s Guide*.

8.3.1.3 Remedy's Hardware Information Form

Detailed hardware information can be provided beyond what can be entered on the Trouble Tickets form by using the Hardware Information form. The User Tools Hardware Information form provides the vehicle to add a description of a hardware problem that corresponds to a trouble ticket. Through this form, the user can enter detailed information about failed hardware components (e.g., part and serial numbers) and the actions taken to correct the problem. This form is accessed by opening RelB-Hardware Information form or via a Hardware Information link from the Trouble Tickets form.

8.3.1.4 Remedy's GUI Notification Tool

The GUI Notification Tool is used as an alternative to email notification to notify the user of a Remedy event. It allows properties and options to be modified via pull-down menus. Examples of GUI notification include a beep, a pop-up window, and a flashing message. In addition, both an email and a GUI notification can be sent if the site so desires.

8.3.2 Submit a Trouble Ticket

When a problem is either found by or reported to User Services, follow the procedure applicable to your system, to create and log trouble tickets. Trouble tickets can be submitted via HTML or via Remedy's user tool – RelB-Trouble Tickets form. Remedy's Contact Log form is used to classify, track, and report contacts of ECS users and operators and also to submit a trouble ticket from a log entry. E-mail is another method of submitting a trouble ticket. The template is available from your System Administrator.

1. For HTML submission:
    a) Access HTML Trouble Ticketing Main page.
    b) Select Submit link, which opens the Submit page.
    c) Fill out the impact, short description, and detailed description fields.
    d) Select Submit.
2. For submission through Remedy:

a) Access Remedy User Tool by logging in to the Remedy host and executing the command **aruser &**.

b) Access the RelB-Trouble Tickets form by selecting **Open** from the **File** menu, highlighting **RelB_TroubleTickets**, and clicking the **New** button.

c) Fill in those fields as specified in Table 8.3.1-1 "RelB-Trouble Ticket Field Description".

d) Select **Save** from the **Actions** menu (or click on the **Save** button).

3. For submission from a Remedy Contact Log entry:

a) Open RelB-Contact Log form.

b) Fill out Contact Log ID and Contact Information. If the contact is a registered Remedy user, the contact information is filled out automatically.

c) Fill in Short Description (limit is 128 characters).

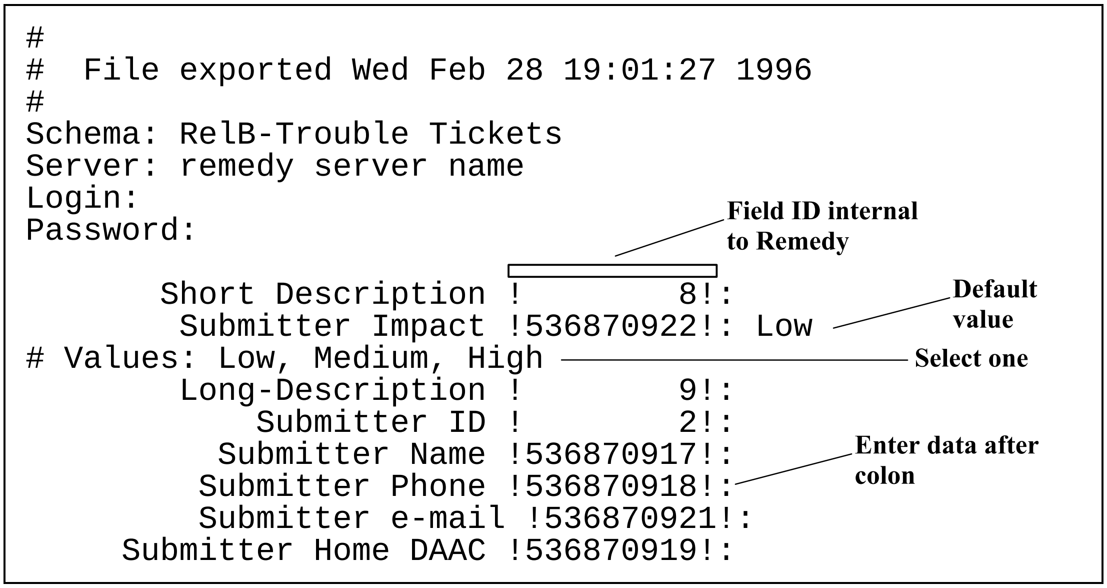
d) Click on **Create TT** button.

4. For submission via E-mail:

a) Obtain Template from your System Administrator.

b) Address the message to arsystem@____________.___________.________.

c) Copy template into message area. DO NOT INCLUDE AS AN ATTACHMENT. DO NOT ALTER TEMPLATE. The template is presented in Figure 8.3.2-1. The # sign indicates comments, which are not read by Remedy. **Enter data as indicated in Figure 8.3.2-1.** Send message.



```
#
#  File exported Wed Feb 28 19:01:27 1996
#
Schema: RelB-Trouble Tickets
Server: remedy server name
Login:
Password:

       Short Description !        8!:
        Submitter Impact !536870922!: Low
# Values: Low, Medium, High
        Long-Description !        9!:
            Submitter ID !        2!:
          Submitter Name !536870917!:
         Submitter Phone !536870918!:
         Submitter e-mail !536870921!:
     Submitter Home DAAC !536870919!:
```

Field ID internal to Remedy
Default value
Select one
Enter data after colon

Figure 8.3.2-1. Trouble Ticket E-mail Template

8.3.3 Reviewing and Modifying Open Trouble Tickets

Trouble tickets may need to be modified based on better understanding of the nature of problems defined and revised resolutions from the Maintenance Engineer investigations, Sustaining Engineering inputs, Developer inputs, Problem Review Board decisions, Change Control Board decisions, and/or Failure Review Board decisions. The results will be factored into revisions and/ or additions to the Trouble Ticket log.

1. For HTML Review and Modification of Trouble Tickets:

   a) Access HTML Trouble Ticketing Main (see Section 8.4). Trouble Tickets can be ***submitted, queried or modified***.

   b) Select List link which opens the List page and shows each Trouble Ticket's Identification, Short Description, and Status.

   c) Select the Trouble Ticket Id to get a more detailed description of that particular Trouble Ticket.

2. For Reviewing and Modifying Trouble Tickets through Remedy:

   a) Access Remedy User Tool (see Section 8.3.2.2a).

   b) Access the RelB-Trouble Tickets form by selecting **Open** from the **File** menu, highlighting **RelB_TroubleTickets**, and clicking the **Search** button.

   c) In the resulting **Modify RelB-TroubleTickets** window, enter a search criterion and then select **Search** from the **Actions** menu (or execute that menu selection with no search criterion entered, to obtain a list of all Trouble Tickets, and then select one to modify).

   d) Review and/or modify the Trouble Ticket fields as necessary.

   e) Select **Save** from the **Actions** menu (or click on the **Save** button).

8.3.4 Forwarding Trouble Tickets

Trouble ticket administrative reports are forwarded for local and system-wide usage. The trouble ticket contains all forwarding information; once forwarded, it goes to the RelB-TT-ForwardToSite holding area (transparent to the user). The RelB-TT-Sites schema is used to indicate the site name and email address to be used in forwarding. To forward a trouble ticket:

1. Access the Trouble Ticket to be forwarded (see 8.3.3a to 8.3.3c).
2. Select **Forwarded** from the pull-down menu in the **Ticket Status** field.
3. Select the TT's destination using the **Forward-to** field pick list.
4. Select the **Forward** button.
5. Select **Save** from the **Actions** menu (or click on the **Save** button).

8.3.5 Adding Users to Remedy

The TT Administrator uses the Remedy User form to grant access to the Remedy tool. Users who leave the ECS program can be deleted. The Remedy *Action Request System 4.5 Concepts Guide*, Chapter 6, “Controlling Access to AR System,” summarizes access control and license elements. There are three classes of licenses that can be assigned to users:

- The **Read** class allows users to submit requests (e.g., Trouble Tickets, User Contact Log records) and modify their own requests. Thus, there are no license restrictions on the number of users who can be granted permission to create and read trouble tickets.
- The **Fixed Write** class provides the capabilities of the Read class plus the ability to modify existing requests submitted by others. A Fixed Write license assigned to a user is always reserved for that user. The AR system administrator must have a Fixed Write license.
- The **Floating Write** class provides the capabilities of the Fixed Write class, but these licenses are not reserved to a single user; instead, they are available on a first-come, first-served basis. Each DAAC has five Floating Write licenses. It is appropriate to assign Floating Write licenses to those who will need to modify Trouble Tickets (e.g., those who will be assigned to work and resolve problems). The limit of five licenses should pose no problem, as it is unlikely that there will be many instances in which more than five people will be trying to modify Trouble Tickets at one time.

The Remedy *Action Request System 4.5 Workflow Administrator’s Guide*, Chapter 3, “Defining Access Control,” provides information on using the **User** form (a "form" represents a table in the Remedy database) to add registered users.

8.3.6 Changing Privileges in Remedy

Changing privileges in Remedy, or controlling privileges of those who have access to Remedy, is done by the CM Administrator. There are numerous Remedy privilege groups for ECS, and a change to the privileges of any group requires an approved Configuration Change Request (CCR). Access privileges determine which forms a user may open and specify whether a user may change information in a field or merely view it.

The Remedy *Action Request System 4.5 Workflow Administrator’s Guide*, Chapter 3, “Defining Access Control,” provides detailed information about access control and privileges. The chapter includes a section on “Adding Groups” that provides instruction on using the **Group** form to create access control groups to which users may then be assigned. A user’s privileges may be changed in two ways:

- changing the group to which the user is assigned.
- changing the access privileges of the group.

The **User** form is used to implement group assignment. To change the access privileges of a group, you use the **Admin** tool. There are two primary ways to change the access privileges of a group. Both ways use the **Group Permissions** window. This window is obtained by selecting **Groups** on the **Remedy Administrator – Server Window** and then double clicking on one of the listed groups. One way to change access privileges is to use the **Form Permissions** tab of the **Group Permissions** window to select which forms are available to members of the group when they access the **Open** dialog box of the **User** form. The second way is to use the **Field Permissions** tab to specify which fields of a form the members of the group may change and which fields they may view but not change.Groups have already been created to accommodate all privileges needed by Remedy Users for Release B. These groups are identified in Table 8.3.6-1.

Table 8.3.6-1. Table of Access Control Groupings

Groups	Description	Access Type
Operator	Submits trouble ticket internally.	Change
User Services	Submits trouble ticket internally for user.	Change
Operations Supervisor	Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change
Resource Manager	Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change
Resolution Technician	Attempts to resolve problem.	Change
Problem Review Board Chair Person	Reviews proposed solutions	Change
Administrator	Adds groups and users. Changes permissions. Sets escalation times. Sets menu items. Etc.	Change
Sub-Administrator	Same functions as Administrator but only with certain Schemas.	Change
Browser	Read only permission.	Read
Customize	Can use all features of the customize facility.	Change
Submitter	Place holder for anyone that submits a trouble ticket.	NA
Assignee	Place holder for anyone that is assigned a trouble ticket.	NA
Public	Read only permission. Guest users are automatically put in this group.	Read
NotifyNewEscal	Everyone that will be notified on an escalation due to trouble ticket being in “New” status.	Read
NotifyAssignedEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Assigned” status.	Read
NotifySolPropEscal	Everyone that will be notified on an escalation due to TT being in “Solution Proposed” status.	Read
NotifyImpSolEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Implement Solution” status.	Read
NotifySolImpEscal	Everyone that will be notified on an escalation due to TT being in “Solution Implemented” status.	Read

8.3.7 Modifying Remedy's Configuration

RelB-Trouble Ticket forms' pulldown menus can be customized. Customization is achieved through the Admin Tool by modifying the RelB-Menu-Closing Codes, RelB-Menu-Hardware Resources, RelB-Menu-Software Resources, RelB-Menu-Key Words, RelB-Menu-Problem Type, or Sites forms.

1. To modify the Remedy environmental variables, refer to the *Action Request System 4.5 Workflow Administrator's Guide*.

   NOTE: No administrative configuration should be made without proper configuration change approval.

8.3.8 Generating Trouble Ticket Reports

A set of predefined reports is available through Remedy. These reports are trouble ticket administrative reports generated for local and system-wide usage. There are several types of predefined reports, including:

- Assigned-to Report – provides a report of the number of Tickets assigned to technicians.
- Average Time to Close TTs – provides a report of the average time to close trouble tickets.
- Hardware Resource Report – provides a report sorted and grouped by Hardware Resources and Closing Codes.
- Number of Tickets by Status – provides the number of Trouble Tickets grouped by Status.
- Number of Tickets by Priority – provides the number of Trouble Tickets grouped by assigned priority.
- Review Board Report – provides a report of the details of TTs for the TT Review Board.
- SMC TT Report – provides a report to be sent to the SMC.
- Software Resource Report – provides a report sorted by Software Resources and their Closing Codes.
- Submitter Report – indicates by submitter the number and type of trouble tickets in the system.
- Ticket Status Report – provides a report sorted and grouped by Ticket Status.
- Ticket Status by Assigned-to – provides a report sorted and grouped by the last person assigned to a Trouble Ticket.

Most of the time, you will probably select a report from the list, using the **Report** window. If you need a custom report, it is possible to select data from the database and create a report using an appropriate separate software application (e.g., a spreadsheet program).

8.4 Using Hypertext Mark-up Language (HTML) Screens

The hypertext mark-up language (HTML) Trouble Ticket Main Screen ("ECS Trouble Ticketing: Menu") provides an introduction on how to use the Trouble Ticketing HTML, and is used by registered ECS users to go to either the Submit page or List page.

Selecting **Submit a Trouble Ticket** will bring up the Trouble Ticketing Submit screen.

Selecting **List the [username] Trouble Tickets** will bring up the Trouble Ticketing List screen.

Help on the Trouble Ticket HTML screens is available by clicking on the Trouble Ticket Help icon at the bottom of the screen .

8.4.1 ECS Trouble Ticketing HTML Submit Screen

The HTML Trouble Ticket Submit screen is used by registered ECS users to submit a Trouble Ticket.

Table 8.4.1-1 below provides a description of the Trouble Ticket HTML Submit Screen fields.

Table 8.4.1-1. Trouble Ticket HTML Submit Screen Field Description

Field Name	Data Type	Size	Entry	Description
ID	character	30	System generated	Submitter Id
Name	character	30	System generated	Submitter Name
E-mail address	character	64	System generated	Submitter E-mail Address
Phone	character	30	System generated	Submitter Phone Number
Home DAAC	character	60	System generated	Submitter Home DAAC
Impact	selection	4	Required	Impact to Submitter
Short description	character	125	Required	Short description of problem
Detailed problem description	character	245	Optional	Long description of problem

When the information is completed, the user can submit the Trouble Ticket by clicking on the **Submit** button on the lower half of the screen. The Problem Information Fields can be cleared by clicking on the **Reset** button. The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.2 ECS Trouble Ticketing HTML Success Screen

The HTML Trouble Ticket Success screen is used by registered ECS users to ensure successful submission and report Trouble Ticket Id.

From this screen, the user is provided with the following information/options:

- Confirmation that the trouble ticket was successfully submitted, the trouble ticket identification number, and who submitted the trouble ticket.
- Notification that an E-mail message has been sent to the user indicating that a Trouble Ticket has been submitted and when it was closed. Selecting this Trouble Ticket will open the Trouble Ticket Detailed Screen.
- Instructions telling the user how to check the progress of Trouble Ticket resolution.

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.3 ECS Trouble Ticketing HTML List Screen

The HTML Trouble Ticket List screen is used by registered ECS users to List Trouble Tickets for a user and links the listed Trouble Ticket Number to the Trouble Ticket Detailed Screen.

Table 8.4.3-1 below provides a description of the Trouble Ticket HTML List Screen fields.

Table 8.4.3-1. Trouble Ticket HTML List Screen Field Description

Field Name	Data Type	Size	Entry	Description
Trouble Ticket Number	character	15	System generated	Trouble Ticket Id
Problem Short Description	character	125	System generated	Short Description of Problem
Status	character	20	System generated	Status of Trouble Ticket

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.4 ECS Trouble Ticketing HTML Detailed Screen

The HTML Trouble Ticket Detailed screen is used by registered ECS users to see a more detailed output of a Trouble Ticket.

Table 8.4.4-1 below provides a description of the Trouble Ticket HTML Detailed Screen fields.

Table 8.4.4-1. Trouble Ticket HTML Detailed Screen Field Description

Field Name	Data Type	Size	Entry	Description
ID	character	30	System generated	Submitter Id
Name	character	30	System generated	Submitter Name
E-mail address	character	64	System generated	Submitter E-mail Address
Phone	character	30	System generated	Submitter Phone Number
Home DAAC	character	60	System generated	Submitter Home DAAC
Status	selection	4	System generated	Status of Trouble Ticket
Impact	selection	4	System generated	Impact to Submitter (low, medium, high)
Short description	character	125	System generated	Short description of problem
Detailed problem description	character	245	System generated	Long description of problem
Log	character	unlim.	System generated	Diary of problem resolution

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.5 ECS Trouble Ticketing HTML Help Screen

The HTML Trouble Ticket Help screen is used by registered ECS users to get help with the HTML screens.

This screen provides general information on the following:

- Index -- links that scroll the screen to the Introduction, Submit Page, and List Page sections listed below.
- Introduction – provides information about the Trouble Ticket Help page
- Menu Page – describes the Trouble Ticketing Menu page.
- Submit Page – describes the Trouble Ticket Submit page.
- Success Page – describes the Trouble Ticket Success page.
- List Page – describes the Trouble Ticket List page.
- Detailed Page - describes the Trouble Ticket Detailed page.

8.5 Emergency Fixes

Emergencies may be in real time with the understanding that the Trouble Ticket System must be brought up-to-date as soon as possible after implementing the repair. The example presented below, involves a hardware failure. The problem needs to be resolved quickly to bring a system back into operation. The resolution requires emergency replacement of a component that is of a later version than is contained in the original equipment. The scenario is summarized in Table 8.5-1.

Scenario — An Example of an Emergency Change Procedure

It is 7:00 on a Saturday evening. The DAAC operator detects a problem with the automated tape library (ATL) and reports the problem to the Trouble Ticket System. The trouble ticket is routed to the System Administrator, who confirms that the system will not operate and notifies the site Maintenance Engineer. After running further diagnostics, the Maintenance Engineer reports the problem and symptoms to the OEM's maintenance desk. The original equipment manufacturer (OEM) maintenance representative arrives and concludes that a controller card has failed. The only card the OEM has immediately available is of a later version and no spares are available on site. It will be Monday at the earliest before a replacement board of the same revision level can be located. The site maintenance engineer reports this to the operations Crew Chief (i.e., shift leader) for a decision.

The DAAC cannot afford to have the ATL down until Monday. The Crew Chief calls the DAAC manager at home, apprises him of the situation, and obtains approval to replace the board with the later version if tests conclude that it works properly. The OEM's maintenance representative installs the board. The site's sustaining engineer tests the new controller board, finds that it works properly, and brings the ATL back on-line. The sustaining engineer updates the trouble ticket to document the configuration change and the authority for the change, and forwards it to the site CMA. The site Maintenance Engineer updates the property record with the model, version, and serial number of the new board.

The site CM Administrator reviews the trouble ticket, and presents it to the local CCB for approval. The CMA then updates the Baseline Manager with the new configuration and TT number authorizing the change. At this point, the site is operational at variance from the system baseline, i.e., site unique, and is at risk of loosing maintenance support from M&O.

The site CMA forwards the trouble ticket to the SMC, presented to the PRB for priority review and is solved or translated into an NCR. The M&O PRB reviews all emergency TTs to assess whether there may be impacts to the ECS and/or applicability to other sites. The M&O CCB monitors all open NCR promotion and approves them for closure.

In the event that it is later discovered that the new version controller board has adverse impacts when operating in the ECS configuration, a board of the original version will have to be obtained to replace the newer version. In such cases, the action will be recorded on a new trouble ticket, citing the previous CCR.

Table 8.5-1 summarizes emergency procedures that might be taken during an after-hours, over-the-weekend emergency hardware failure.

Table 8.5-1. Example of Emergency Change Procedure

Operator/User	System
Operator prepares trouble ticket to report ATL controller failure.	Trouble ticket recorded.
System Administrator and Maintenance Engineer confirm ATL controller failure, call ATL maintenance vendor, report call and time in Trouble ticket.	Diagnosis and vendor call recorded in trouble ticket.
Maintenance vendor isolates failure to the controller card. The later version card is the only card available.	
Crew Chief notified of situation and decision needed to bring ATL up to full operating capability. Approves use of the newer version card, records decision in the trouble ticket, forwards trouble ticket to Sustaining Engineer.	
Maintenance vendor installs card, tests using hardware diagnostics. Crew Chief authorizes controller to be brought back on-line.	
Maintenance Engineer records card installation by model/version into the trouble ticket.	Trouble ticket action recorded.
Sustaining Engineer reads trouble ticket and prepares for discussion at 8:30 am meeting. Updates the TT.	Install action recorded in TT. TT routed to the CMA.
CMA updates site baseline, forwards TT to the CCB. When CCB approves the action, CMA forwards to SMC.	Site ATL baseline updated in Baseline Manager.
M&O reviews emergency NCR, checks for applicability to other sites, opens new NCR if other sites require change.	

This page intentionally left blank.

9. Configuration Management Procedures

The procedures that have been prepared are applicable to all of ECS including all hardware, software, and firmware components of systems or subsystems developed or acquired by the ECS contract and/or delegated to configuration management control by the operational site-level organizations. The procedures are applicable to all items maintained by the ECS Sustaining Engineering Organization in support of ECS mission-specific projects and multiple mission-specific institutional facilities. The procedures are not applicable to those entities controlled by higher level ESDIS Project Office CM Plans. The procedures also apply to ESDIS authorized replacements of or augmentations to fungible assets at extant facilities. CM procedures already in place may be used by the contractor subject to direction from the Change Control Board (CCB) chair person.

Some major features of the approach being offered here include:

- Customers participate in establishing the procedures;
- The M&O CCB performs a support role for ESDIS and its designated on-site CCBs by processing system-level CCRs & Trouble Tickets;
- Prioritization, automated tools, and procedures are used for handling change requests;
- Diverse/Strategic representation at hierarchical CCBs facilitate a path for speedy escalation/resolution of problems/issues;
- Local organizations have the needed autonomy to accomplish their mission with the minimum necessary outside intervention to promote timely resolution of local problems and enable timely production of data products;
- Proper use and deployment of CM database assets to support all CCBs allows management monitoring, control, and analysis of activities;
- Coordination with the Failure Review Board allows coordinated response to problems and filtering of prioritized issues; and
- Common CM tools will be used in all elements of the ECS Project during operations.

The procedures are organized into nine major sections that address the flow-down of procedures from the ECS system-level to the site-level with references to site-tailoring of procedures where applicable. The topics include (Section 9.1) configuration identification, (Section 9.2) change control processes, (Section 9.3) configuration status accounting, (Section 9.4) configuration audits, (Section 9.5) data management, and the use of standardized CM tools known as (Sections 9.6 and 9.7) Software CM Manager (ClearCase), (Section 9.8) Change Request Manager (Distributed Defect Tracking System), and (Section 9.9) Baseline Manager (ClearCase).

9.1 Configuration Identification Procedure

9.1.1 Purpose

The purpose of configuration identification during maintenance and operations is to incrementally establish and maintain the definitive basis of control and status accounting for the ECS control items. To accomplish configuration identification for both hardware and software, the configuration management (CM) administrator (CMA) shall ensure the maintenance of each ECS configuration controlled item in an operational baseline by executing the following tasks:

a. Assign identifiers to configuration items (CIs) and their component parts and associated configuration documentation, including revision and version number where appropriate. Assign serial and lot numbers, as necessary, to establish the CI effectivity of each configuration of each item of hardware and software.

b. Follow ECS developer guidelines as referenced below in Section 9.1.3.

c. Follow vendor nomenclature for COTS items.

d. Apply operation and maintenance (O&M) version name extensions to ECS modified item nomenclature following the rules in Section 9.1.4.

e. Follow author-designated version control and nomenclature for documents and follow guidelines from the ECS Librarian (cf. Section 20, *Library Administration*)

f. Support the ECS Librarian's efforts to maintain linkage of the ECS documentation to ECS configuration items in the Baseline Manager tool. Ensure that the marking and labeling of items and documentation with their applicable identifiers enables correlation between the item, configuration documentation, and other associated data.

g. Maintain a release system for configuration changes (cf. Section 9.2, *Configuration Change Control Procedures)*.

h. Maintain views of operational baselines using the Baseline Manager tool (cf. Section 9.9).

i. Ensure that applicable identifiers are embedded in the source and object code.

9.1.2 Applicable to

All ECS CM Administrators and support personnel.

9.1.3 References

ESDIS CM Plan

Configuration Management Plan for the Science Data Processing Segment of the ECS Project	102-CD-003-004
ECS Software Build Process	CM-1-045
Embedded Versioning	CM-1-031

Data Identification Numbering	DM-1-002
DoD Mil-Std-973	

9.1.4 Procedures

9.1.4.1 Extended Configuration Identification

The extended configuration identification for ECS is of the standard form:

Control Item.Release.Organization.#_Dev.#_M&O.#_center

where:

- *Control Item* is the ECS Project designation of the CI at RRR turnover. The CI naming convention has been explained in CM-1-045, *ECS Software Build Process*.
- *Release* is the major release, A, B, C, or D.
- *Organization* is the organization that established the configuration. Legal values are DEV for development, M&O for Maintenance and Operations system-wide, or center (e.g., SMC, EOC, EDC, GSFC, etc.) for center unique.
- *#_Dev* is a numeric identifier applied by the development organization to the major release and/or a minor release.
- *#_M&O* is a numeric identifier applied by the M&O/SEO organization. This field is used by the SEO organization to establish the system M&O baseline.
- *#_Center* is a numeric identifier applied by each center. This field is used by the operational centers to establish the site specific baseline.

For example, at the TRMM Development Release RRR, the ECS CCB establishes the initial operational baseline. Assume this baseline is identified as CI.A.DEV.3. CI.A.DEV.3 is delivered to the ESDIS CCB. After ESDIS CCB acceptance, the M&O organization will configure and build its first, system-wide baseline, CI.A.M&O.3.0. If it is assumed that some M&O tailoring is applied, the baseline released to the operational centers is CI.A.M&O.3.1. Each center then configures a center specific baseline. The RRR baseline for EDC, GSFC, LaRC, and NSIDC as well as the SMC and EOC is built from CI.A.M&O.3.1 and are identified as CI.A.EDC.3.1.1, CI.A.GSFC.3.1.1, CI.A.LaRC.3.1.1, CI.A.NSIDC.3.1.1, CI.A.SMC.3.1.1 and CI.A.EOC.3.1.1.

9.1.4.2 Other Procedures as Applicable

The CM Administrator will author other configuration identification procedures applicable to the local site environment to carry out the objectives listed in the Section 9.1.1 Purpose.

9.2 Configuration Change Control Procedures

9.2.1 Purpose

The ESDIS CCB chartered Change Control Boards (CCBs) shall apply configuration control measures to all the ECS configuration items and the associated documentation prior to the time it is baselined for operations. The CCBs shall apply configuration control measures to

a. Ensure effective control of all CIs and their approved documentation;

b. Provide effective means, as applicable, for (1) proposing engineering changes to CIs, (2) requesting deviations and waivers pertaining to such items, (3) preparing notices of revision, and (4) preparing Specification Change Notices; and

c. Ensure the implementation of approved changes.

9.2.2 Applicable to

All ESDIS chartered ECS CCBs.

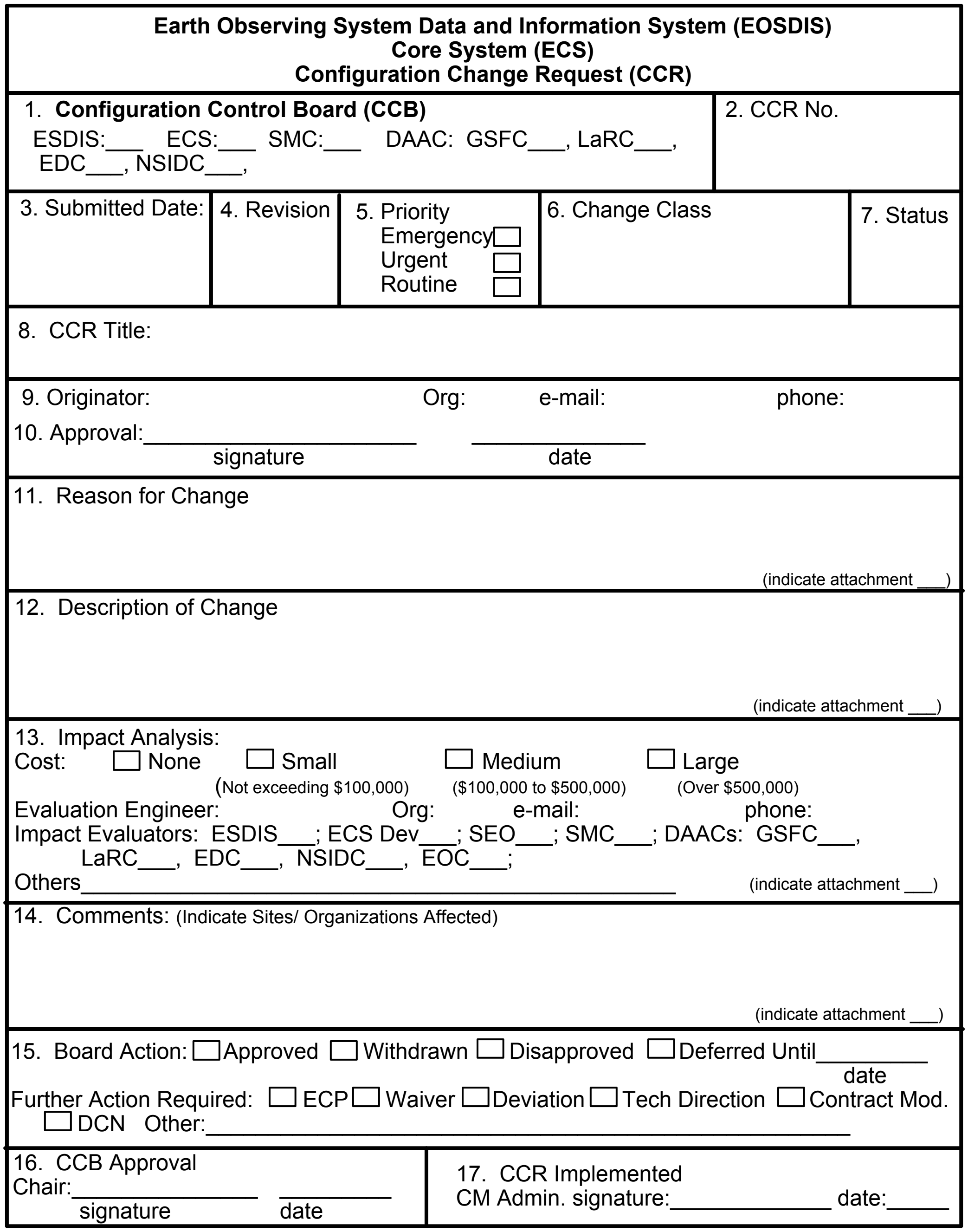
9.2.3 References

ESDIS CM Plan	
MO&DSD CM Plan	
Configuration Management Plan for the Science Data Processing Segment of the ECS Project	102-CD-003-004
CCB Change Control Process	CM-1-004

9.2.4 Procedures

9.2.4.1 Configuration Change Request Preparation

The Configuration Change Request (CCR) form in Figure 9.2.4-1 has been developed as a medium for the drafting of CCRs throughout the ECS Maintenance and Operations environment for changes processed locally at ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, EDC, NSIDC, and ORNL).. For changes processed by the ESDIS Change Control Board (CCB), use ESDIS own forms. For ECS CCBs, use the form available at the URL http://dmserver.gsfc.nasa.gov/forms/formindex.html. There are numbered items on the form that correspond exactly to the data entry required to be performed by the respective Configuration Management Administrator who maintains CCR records for the CCB on the distributed implementation of the Change Request Manager tool. Each CCB will have unique CCR identification sequence numbers. Each CCB can forward CCRs and reports from the Change Request Manager to SMC where SEO processes system-level CCRs for ESDIS CCB. The ESDIS CM Plan will determine the charter of the respective CCBs and thus the scope of CCR issues to be addressed by the site CCBs.



Earth Observing System Data and Information System (EOSDIS)
Core System (ECS)
Configuration Change Request (CCR)

1. **Configuration Control Board (CCB)**
ESDIS:___ ECS:___ SMC:___ DAAC: GSFC___, LaRC___, EDC___, NSIDC___,

2. CCR No.

3. Submitted Date:

4. Revision

5. Priority
Emergency □
Urgent □
Routine □

6. Change Class

7. Status

8. CCR Title:

9. Originator: Org: e-mail: phone:

10. Approval:______________________ ______________
signature date

11. Reason for Change

(indicate attachment ___)

12. Description of Change

(indicate attachment ___)

13. Impact Analysis:
Cost: □ None □ Small □ Medium □ Large
(Not exceeding $100,000) ($100,000 to $500,000) (Over $500,000)
Evaluation Engineer: Org: e-mail: phone:
Impact Evaluators: ESDIS___; ECS Dev___; SEO___; SMC___; DAACs: GSFC___, LaRC___, EDC___, NSIDC___, EOC___;
Others____ (indicate attachment ___)

14. Comments: (Indicate Sites/ Organizations Affected)

(indicate attachment ___)

15. Board Action: □Approved □ Withdrawn □ Disapproved □ Deferred Until_________
date
Further Action Required: □ ECP □ Waiver □Deviation □Tech Direction □Contract Mod.
□ DCN Other:____

16. CCB Approval
Chair:_______________ _________
signature date

17. CCR Implemented
CM Admin. signature:_____________ date:_____

Figure 9.2.4-1. Configuration Change Request (CCR) Form

The following enumerated text corresponds to the numbered items on the CCR form:

(1) **Change Control Board (CCB) --** The designated CCB is checked-off for changes processed by the ESDIS Change Control Board (CCB) and its ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, EDC, and NSIDC).

(2) **CCR Number --** The unique serialized CCR number is applied at each site.

(3) **Submitted Date --** The date that the CCR was prepared is documented.

(4) **Revision --** The current revision is designated for tracking changed versions of original CCR.

(5) **Priority --** The priority level of the CCR is assigned. Emergency CCRs may have already been implemented on a temporary basis by the Trouble Ticket Review Board (TTRB) with concurrence from the CCB Chair who later receives the CCR to document /implement the permanent change. Urgent items will be reviewed by the next CCB meeting. Routine items will be reviewed as soon as the schedule permits.

(6) **Change Class --** Change Classes are either I or II. Class I will be handled by ESDIS-only because of cost, schedule, and/or mission impacts that may require requirements changes. Class II items do not affect mission requirements, but may have cost and/or schedule implications which affect maintenance, operations, procedures, documentation, site-tailored items, COTS implementation, site installations of core system changes, science SW changes, etc.

(7) **Status --** The following table is a summary of the CCR states maintained by the Distributed Defect Tracking System (DDTS) application SW database tool that implements the Change Request Manager. Note that the hard copy form will not be updated chronically, but will be kept in the master suspense file of the CM Administrator until closed-out with a stamp (item #7 & 15) and appropriate signatures (viz., items 16, & 17).

State Table Composition in DDTS format

State Code	Available States	State Assigned
S	Submit	Submitted
N	New	New
A	Assign-Eval	Assign-Eval
O	Assign-Implement	Assigned-Implement
R	Implement	Implemented
T	Assign-Verify	Assigned-Verify
V	Verify	Verified
C	Close	Closed
D	Duplicate	Duplicate
F	Forward	Forwarded
P	Defer	Deferred

Explanations of DDTS' State Table Content:

Uppercase character in the 1st column is the character stored in the change request record to indicate what state a change request is in. DDTS uses this character to go into the table and extract the descriptive name for display in reports.

Names in the second column is the state in the present tense. They are shown on the DDTS list of states that are available for selection during input. It's also used by some of DDTS' query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Names in the third column is the state in the past tense. They are shown on the DDTS change request record. It's also used by some of DDTS query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Definitions (a state is the stage that a proposed change has reached in its life cycle.):

New - the initial state for all newly entered change requests.

Assign-Eval- state entered when the change request is being assigned to an engineer for evaluation/analysis.

Assign-Implement- state entered when the change request is being assigned to an engineer for development.

Implement-state entered when the proposed change has been developed.

Assign-Verify-state entered when the developed change is being assigned to an engineer for verification testing.

Verify- state entered when a developed change has been tested and verified that it functions properly.

Close- state entered when all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity.

Duplicate-state entered when a change request is determine to be duplicate of an existing change request. Duplicate change request identifies change request being duplicated.

Forward-state entered when a change request needs to be forwarded to another DDTS defined project (In DDTS terminology, a project is a grouping of change requests. For example, a change request from a site project can be forwarded to an ECS project.).

Defer- state entered when activity on a change proposal has to be postponed.

(8) **CCR Title --** The CCR title is supplied by the originator.

(9) **Originator --** The originator name, organization, e-mail address, and phone number is given.

(10) **Approval --** The CCR is approved by the designated management authority which is assigned by the CCB. This sponsorship requirement acts as a primary filter to eliminate from consideration those CCRs that cannot be implemented or which have no ECS site management support.

(11) **Reason for Change --** The reason for the change is narrated on the form and/or the designated attachment.

(12) **Description of Change --** The proposed implementation of the change is narrated along with any known impacts, resources, and expenses to be incurred.

(13) **Impact Analysis --** Impact analysis is documented in the form of figure 9.2.4-2. The impact analysis is collected by the CCB Chair appointed Evaluation Engineer in coordination with the CM Administrator who maintains the CM records and assembles the review package for the CCB. The Evaluation Engineer documents the list of Impact Evaluators and derives and/ or verifies cost, technical, and schedule impact of the proposed change based on all inputs received. The results of the coordinated CCR Impact Analysis inputs are presented in the CCR Impact Summary form shown in figure 9.2.4-3 part of the CCR review package.

(14) **Comments** -- Comments are added to the CCR to summarize sites and/ or organizations affected by the CCR. Additional comments may address proposed CCB dispositions and recommendations to be indicated by resolutions in item #15.

(15) **Board Action --** CCB actions and follow-up actions that will be facilitated and tracked by the CM Administrator are indicated. Possible CCB dispositions are given as approved, withdrawn, disapproved, and deferred (pending follow-up activities by the indicated schedule date). Further actions are indicated as

Engineering Change Proposal (ECP)-- changed scope of contract requirements

Waiver-- declaration that certain contract requirements no longer apply

Deviation-- change of contract terms or substitution of terms or deliverable requirements

Technical Direction-- order by Contracting Officer's Technical Representative (COTR) to perform certain tasks within the scope of the contract

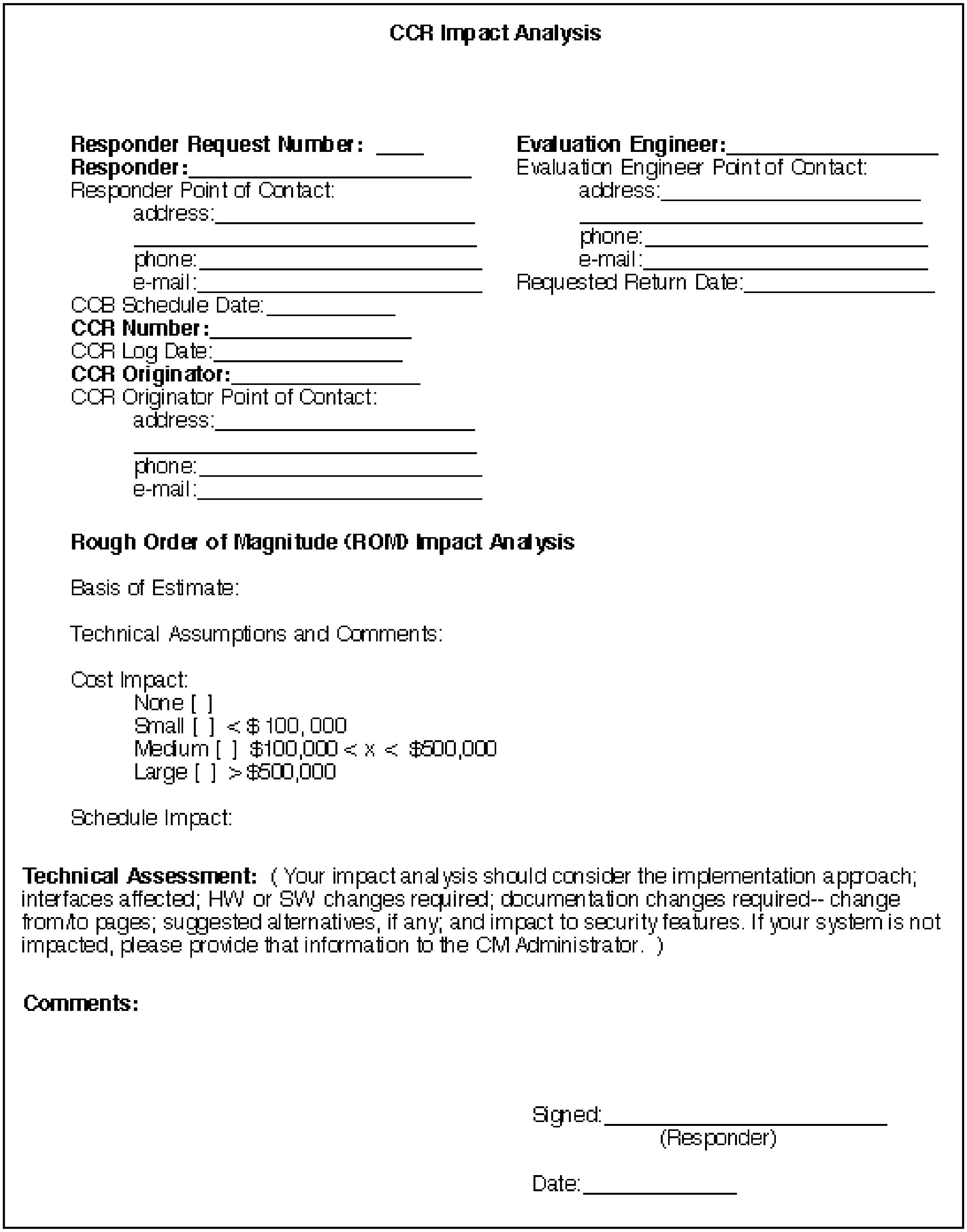
Contract Modification-- changes to the terms of a contract

Document Change Notice (DCN)-- notification of changes to published documents

Others-- Engineering Change Notice, Change Order, Escalate to higher CCB authority, etc.

(16) **CCB Approval --** CCB approval signature authority by CCB Chair or designate.

(17) **CCR Implemented --** This signature and close-out stamp (item #7) are executed by the CM Administrator witnessing the completion of the CCR implementation process which is tracked in the Change Request Manager automated tool DDTS and updated in Baseline Manager for affected version control status changes.



CCR Impact Analysis

Responder Request Number: ____
Responder:________________________
Responder Point of Contact:
address:______________________
____
phone:________________________
e-mail:________________________
CCB Schedule Date:__________
CCR Number:_______________
CCR Log Date:_______________
CCR Originator:______________
CCR Originator Point of Contact:
address:______________________
____
phone:________________________
e-mail:________________________

Evaluation Engineer:________________
Evaluation Engineer Point of Contact:
address:______________________
____
phone:________________________
e-mail:________________________
Requested Return Date:_______________

Rough Order of Magnitude (ROM) Impact Analysis

Basis of Estimate:

Technical Assumptions and Comments:

Cost Impact:
None [ ]
Small [ ] < $ 100, 000
Medium [ ] $100,000 < x < $500,000
Large [ ] > $500,000

Schedule Impact:

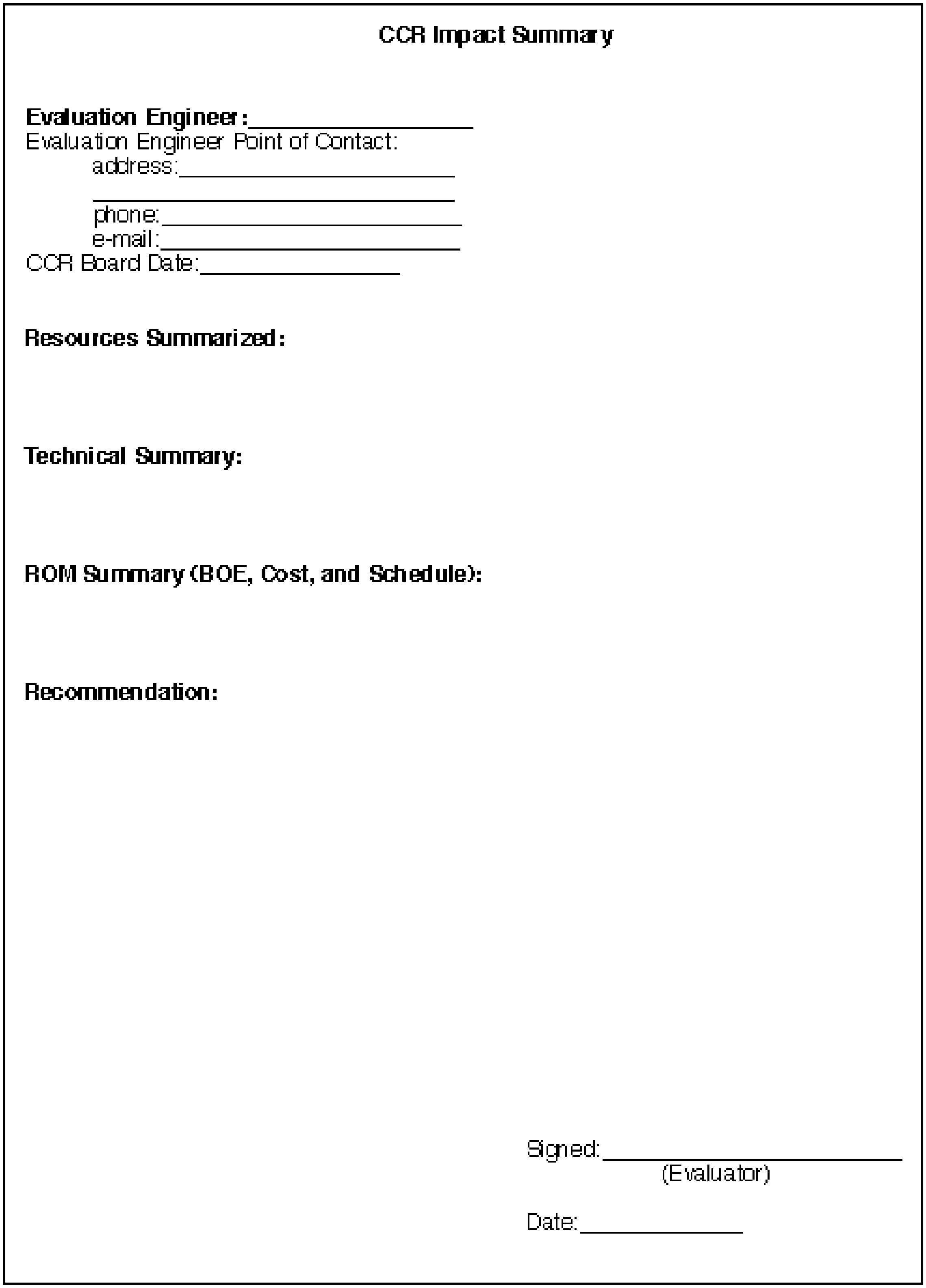
Technical Assessment: (Your impact analysis should consider the implementation approach; interfaces affected; HW or SW changes required; documentation changes required-- change from/to pages; suggested alternatives, if any; and impact to security features. If your system is not impacted, please provide that information to the CM Administrator.)

Comments:

Signed:________________________
(Responder)

Date:_____________

Figure 9.2.4-2. ECS CCR Impact Analysis



CCR Impact Summary

Evaluation Engineer:__________________

Evaluation Engineer Point of Contact:

address:______________________

____

phone:_________________________

e-mail:_________________________

CCR Board Date:________________

Resources Summarized:

Technical Summary:

ROM Summary (BOE, Cost, and Schedule):

Recommendation:

Signed:________________________

(Evaluator)

Date:_____________

Figure 9.2.4-3. ECS M&O CCR Impact Summary

9.2.4.2 Change Control Board Process (System and Site-level CCBs)

The ECS M&O organization will provide administrative and technical support services for the CCB at each site. Each site's CCB is controlled by the host site organization and provides the authority and direction for the ECS contractor to modify the operational baseline. The ESDIS CCB has chartered an ECS Review Board to coordinate ECS system-level changes and problem management via the Sustaining Engineering Organization (SEO) contractor and on-site Review Boards that also act as site CCBs. This is illustrated using the CM Administrator’s workflow for the SEO support of the ECS Review Board in Figure 9.2.4-4 and the On-Site CM Administrator’s workflow for SEO support of the on-site CCB in Figure 9.2.4-5. The problem management process was discussed in detail in Section 8 of this document. Both diagrams illustrate the flow of CCRs through the respective CCBs with inputs from the review boards and evaluators that determine the disposition of proposed changes. Details of this process are given below:

System-level Change Control Procedures

(The enumeration corresponds to the diagram of Figure 9.2.4-4)

(1) Configuration Change Requests are received by the SEO CM Administrator from all sources with regard to the operational ECS Core System as described in Section 9.2.4.1. These changes designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Trouble Ticket Review Board (TTRB) and/ or Failure Review Board as explained below.

(2) Proposed common baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see Section 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The SEO CM Administrator is responsible for logging the CCR into the Change Request Manager (DDTS tool) as described in Section 9.8.

(4) The CCB chair assigns an evaluator and the SEO CM Administrator coordinates impact assessment.

(5) Class I change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the ESDIS CCB for consideration with recommendations from the ECS Review Board.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by ECS Review Board deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of ECS Review Board deliberations are factored into review board resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the SEO CM Administrator to the support activities, i.e., site CCBs, support personnel (SEO), vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the SEO CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The SEO CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features (see section 9.8).

(12) New versions and/or maintenance updates are annotated in Baseline Manager at the SMC and at the affected sites by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at SMC and at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the ECS Review Board.

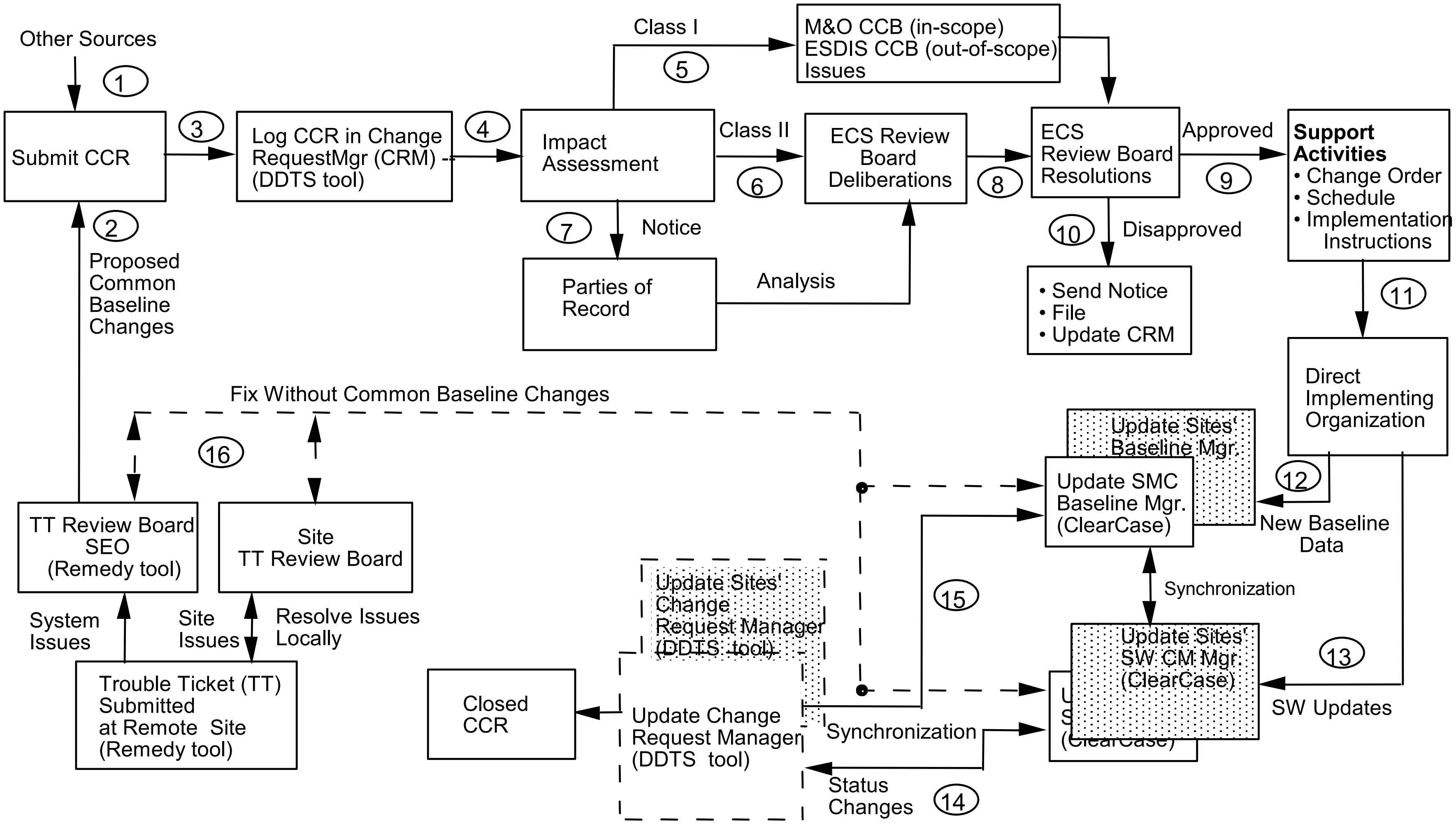

Other Sources
1
Submit CCR
2
Proposed Common Baseline Changes
3
Log CCR in Change RequestMgr (CRM) (DDTS tool)
4
Impact Assessment
Class I
5
M&O CCB (in-scope)
ESDIS CCB (out-of-scope)
Issues
Class II
6
ECS Review Board Deliberations
7
Notice
Parties of Record
Analysis
8
ECS Review Board Resolutions
Approved
9
Support Activities
• Change Order
• Schedule
• Implementation Instructions
10
Disapproved
• Send Notice
• File
• Update CRM
11
Direct Implementing Organization
Fix Without Common Baseline Changes
16
Update Sites' Baseline Mgr.
Update SMC Baseline Mgr. (ClearCase)
12
New Baseline Data
Synchronization
TT Review Board SEO (Remedy tool)
Site TT Review Board
System Issues
Site Issues
Resolve Issues Locally
Trouble Ticket (TT) Submitted at Remote Site (Remedy tool)
Update Sites' Change Request Manager (DDTS tool)
15
Update Sites' SW CM Mgr. (ClearCase)
13
SW Updates
(ClearCase)
Closed CCR
Update Change Request Manager (DDTS tool)
Synchronization
Status Changes
14

Figure 9.2.4-4. Work-Flow Diagram for SEO CM Administrator

Site-level Change Control Procedures

(The enumeration corresponds to the diagram of Figure 9.2.4-5)

(1) Configuration Change Requests are received by the Site CM Administrator from all sources with regard to the **site unique extensions** to the operational ECS Core System as described in section 9.2.4.1. These changes designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Site/ SEO Trouble Ticket Review Board (TTRB) and/or Failure Review Board as explained below.

(2) Proposed site baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see section 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/

emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The Site CM Administrator is responsible for logging the CCR into the Change Request Manager (DDTS tool) as described in Section 9.8.

(4) The CCB chair assigns an evaluator and the Site CM Administrator coordinates impact assessment.

(5) Class I/System Issues change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the ECS Review Board for consideration with recommendations from the Site CCB. Class I issues are further forwarded with recommendations by the ECS Review Board to the M&O CCB for in-scope issues and to the ESDIS CCB for consideration of out-of scope issues with respect to the SOW of the ECS Contract.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by Site CCB deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of Site CCB deliberations are factored into CCB resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the Site CM Administrator to the support activities, i.e., other CCBs, support personnel (SEO), vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the Site CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The Site CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features (see Section 9.8).

(12) New versions and/ or maintenance updates are annotated in Baseline Manager at the affected sites and the SMC by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The on-site TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the ECS Review Board.

Each site's CCB accepts initial release or updates from the ESDIS CCB. Similarly, the Distributed Active Archive Center (DAAC) CCBs will accept product generation software from an ESDIS authority. Local tailoring and installation decisions are determined by the site CCB.

In the case of Evaluation Package (EP) and Prototype deliveries, the ECS CCB as directed by ESDIS will provide a configured, documented, executable with supporting files. Again, installation decisions are determined by the site CCB.

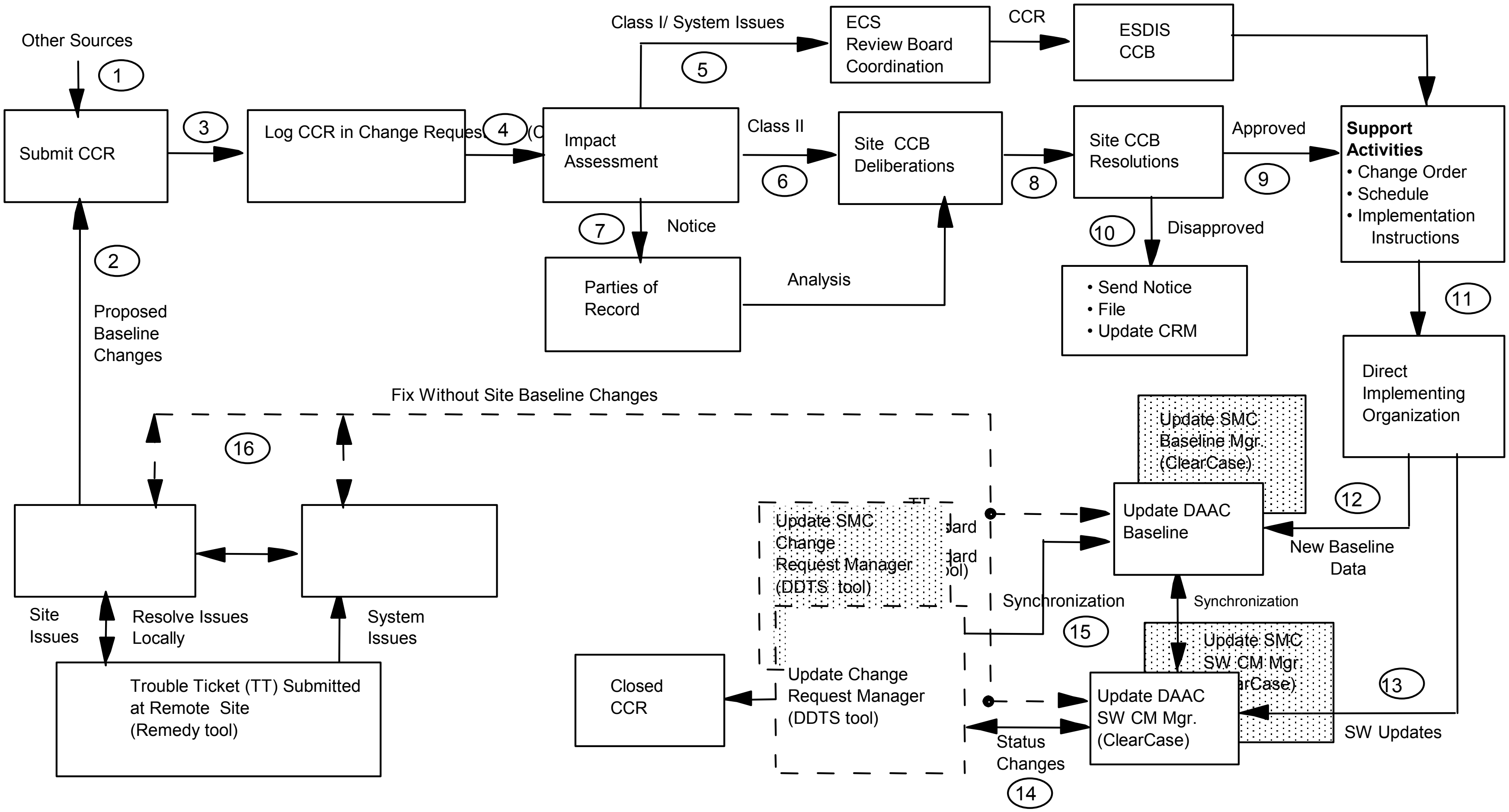


Figure 9.2.4-5. Work-Flow Diagram for Site-level CM Administrator

Each Science Computing Facility (SCF) is assumed to have a configuration control function. For commonality with other sites, it is assumed that this function will be performed by a CCB. A major difference, though, is the ECS contractor does not have an active role in the support of this CCB.

The SCF CCB will provide two types of configuration control:

(1) Configuration control of software and databases that are to be executed in another site's environment. In this mode, it operates very much as the ECS CCB does to establish a product baseline.

(2) Configuration control of SCF resources that are made available to the EOSDIS community. In this mode, its functions are the same as a DAAC CCB.

The ECS M&O CM function at each DAAC will accept science SW and data items from the SCF CCB. These items will be incorporated into the DAAC's operational baseline as directed by the DAAC CCB.

The EOC CCB will control the operational configuration of the required EOC operational baseline. ECS M&O CM will provide services as directed by the CCB.

The ECS Review Board will be charged with the responsibility for centralized coordination and control of ECS CM activities to ensure:

- ECS integrity and quality of service
- Successful coordination with both internal and external networks, systems, and on-site facilities
- Timely ESDIS CCB visibility into and oversight of ECS operations
- Convenient user administrative services

9.2.4.3 Configuration Control - Deviation and Waivers

1. Prior to completion of software development or to purchase of equipment or software, a Deviation from the specification may be granted by the ESDIS CCB.
2. Subsequent to the completion of development or delivery of equipment or software, a Waiver of specific requirements may be granted in order to accept defined nonconforming items. The waiver is traceable to a nonconformance report. A waiver is limited: additional deliveries of like items must conform to approved requirements .
3. Departures from expected configurations that are not at variance with customer-approved requirements are handled as Nonconformance Reports.
4. A request for a Deviation or Waiver consists of a Deviation/Waiver Form shown in figure 9.2.4-6 attached to a CCR form .

5. Instructions for completing the form are listed below. Additional pages should be attached as necessary.

   Dev or Wai: Check the applicable box in accordance with the definitions given on page 1 of this Instruction.

   Waiver Number: Assigned by ECS CM Administrator

   Title: Enter a brief descriptive title. The title should be a statement, e.g., "Accept x in lieu of y".

   Reason: Describe the reason for the deviation or waiver. This may also include the history of the problem and consequences of not implementing the deviation/waiver.

   Existing Requirement: Specify and describe the baseline from which the deviation or waiver departs.

   Departure: Give instructions for the deviation or waiver with reference to the requirement.

   Implementation Scope: Include as appropriate, configuration item number, model no., supplier, subcontractor, series, serial numbers, order no., location, release number, quantity, time period or other criteria delimiting the deviation or waiver.

   Documents Affected: List current release number(s) of affected document(s).

6. The Deviation or Waiver CCR is prepared in accordance with the instructions in section 9.2.4.1 with the following exceptions:

   Change Class: All deviations and waivers are change class I because they depart from approved requirements and must be approved by the ESDIS CCB.

   Description (Title): The title of the CCR will be the deviation or waiver title.

   Proposed Solution: Enter "See Attached Deviation/Waiver form ."

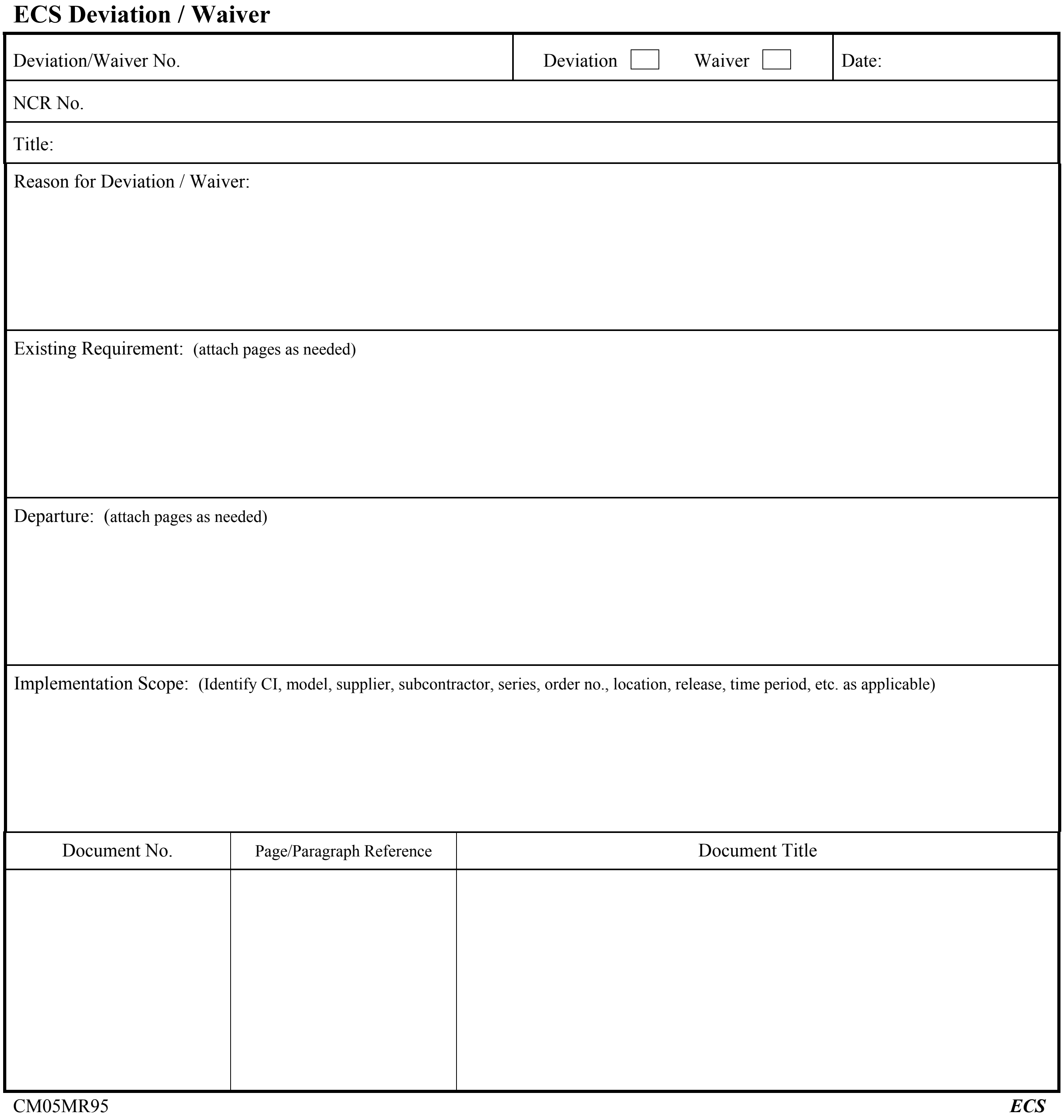
7. Waiver and Deviation CCRs are submitted to the ECS CCB. Upon ECS CCB approval, the CM Administrator forwards the CCR to ESDIS CCB for authorization to implement the deviation or waiver.

8. When the deviation or waiver is authorized, the ECS CM Administrator immediately distributes the authorization information to the appropriate implementors and issues a document change order (DCO) to the Document Management Organization.

9. The SEO Librarian copies the implementation instructions into the List of Deviations and Waivers at the front of the document and inserts the Deviation/Waiver number and effectivity at the point of applicability within the document.

10. Approved deviations/waivers are published via Document Change Notices (DCNs). Adding the deviation/waiver information to the document makes its status general knowledge. However, the deviation or waiver is in effect as soon as it is authorized by the customer .

11. A change in scope, effectivity or closeout, or any other change in a deviation or waiver requires a new CCR. The closeout or change is applied to the document via a Document Change Order in the same procedure as given in paragraphs 7 through 10.



ECS Deviation / Waiver

Deviation/Waiver No.	Deviation □ Waiver □	Date:
NCR No.		
Title:		
Reason for Deviation / Waiver:		
Existing Requirement: (attach pages as needed)		
Departure: (attach pages as needed)		
Implementation Scope: (Identify CI, model, supplier, subcontractor, series, order no., location, release, time period, etc. as applicable)		

Document No.	Page/Paragraph Reference	Document Title

CM05MR95 *ECS*

Figure 9.2.4-6. Deviation/Waiver Form

9.3 Configuration Status Accounting Procedures

9.3.1 Purpose

Operational phase configuration status accounting (CSA) consists of recording and reporting information about the configuration status of the ECS Project's documentation, hardware and software throughout the Project life cycle. Periodic and ad hoc reports keep ESDIS informed of configuration status as the operational mission evolves. Reports to support reviews and audits will be extracted as needed starting from the RRR.

The Baseline Manager tool described in section 9.9 records and tracks as-built products designated as ECS control items (i.e., custom, COTS, science, toolkits, etc. SW and HW items along with their associated documentation and records) and historical versions of ECS operational configurations. Baseline Manager, which is updated with the acceptance tested version of the ECS baseline at RRR, records and reports M&O document change status and histories, mission milestone baselines, and change status.

CSA entails maintaining version histories of delivered and maintained products as well as histories of operational baselines and changes made to each baseline. Additionally, CSA tracks the status of proposed changes from initial CCR submission to ultimate disposition and/or implementation. CSA also maintains historical records of CCRs.

9.3.2 Applicable to

All ESDIS chartered CCBs.

9.3.3 References

ESDIS CM Plan

Configuration Management Plan for the Science Data Processing Segment of the ECS Project	102-CD-003-004

9.3.4 Procedures

The following are topical items subject to periodic or ad hoc reporting on behalf of the respective CCB or a system-level summary of information that will be reported by the SEO CM Administrator representing the operational baseline for all the ECS sites.

(a) **New CCRs and Revisions.** This is a standard Change Request Manager report (cf. Section 9.8). This report will be issued monthly and summarized annually.

(b) **CCB Review.** Distribute CCR copies for review (and Impact Analysis forms if applicable). Print the agenda and distribute prior to the meeting.

(c) **Open Action Items.** Open action items should be statused regularly between meetings.

(d) **CCB Meeting.** Record the CCB’s disposition of each CCR.

(e) **Record Action Items.** Record actions, assignments, and due dates.

(f) **SEO Librarian Maintained Document Changes.** When all authorized document changes have been accomplished prepare DCN, post the final version on the ECS Document Data Server and distribute hardcopy as required.

(g) **Minutes Distribution.** Distribute minutes to the standard distribution and inform actionees of assigned action items.

(h) **CCR Implementation Status.**

- After CCB disposition, regularly status open CCRs until closure.
- Class I events include: CCR to ECS Review Board for review/appoval; Technical Review Board; and ESDIS Disposition
- Further events are as follows for M&O implementation status: Consent Obtained; Item Received; Installed; Document Completed; etc.
- CCR CLOSED: A Class I CCR is not closed until the ESDIS contract officer's authorization is received or the reference CCR has been withdrawn.
- Class II document change CCRs may be closed with the CM Administrator's issuance of the DCN.
- Other non-document change CCRs may be closed when the originator verifies to the CM Administrator that all specified changes have been implemented.

9.4 Configuration Audits

9.4.1 Purpose

SEO will support Functional Configuration Audit /Physical Configuration Audit (FCA/PCA) by IATO at RRR. SEO will also support audits by ESDIS and our own Quality Office functions. Internal CM self-audits will be conducted by the SEO. Self-audits evaluate the Project's compliance with the EOS Configuration Management Plan and the ESDIS CMP. The CM self-audits will verify:

- That CM policies, procedures, and practices are being followed.
- That approved changes to documentation, and to software and hardware products are properly implemented.
- That the as-built documentation of each CI agrees with the as-deployed configuration or that adequate records of differences are available at all times.

A post-audit report is written outlining the specific items audited, audit findings, and corrective actions to be taken. All action items are tracked to closure.

In addition, SEO supports formal audits scheduled and conducted by ESDIS. These audits are conducted to validate that each ECS CI is in conformance with its functional and performance requirements defined in the technical documentation. The audits validate that:

- The as-built configuration compares directly with the documented configuration identification represented by the detailed CI specifications.
- Test results verify that each ECS product meets its specified performance requirements to the extent determinable by testing.
- The as-built configuration being shipped compares with the final tested configuration. Any differences between the audited configuration and the final tested configuration are documented.
- When not verified by test, the compatibility of ECS products with interfacing products or equipment is established by comparison of documentation with the interface specifications which apply.
- COTS products are included in audits as integral parts of the ECS baseline.

9.4.2 Applicable to

All ESDIS chartered CCBs.

9.4.3 References

ESDIS CM Plan

Configuration Management Plan for the Science Data Processing
Segment of the ECS Project 102-CD-003-004

9.4.4 Procedures

The audits will be standardized for a limited set of issues that drive the process for which the audit is taken, viz., FCA/PCA, Security Issues, General Accounting, Test Readiness Review, or Operational Certifications. The documented basis for the audit process will be maintained in the Baseline Manager CM tool (cf. Section 9.9). Alternatively, the Version Description Document (VDD) will be used to document auditable changes to configured articles that are issued at the ECS configuration item (CI) level. The use of the VDD is discussed in 102-CD-003-004, *Configuration Management Plan for the Science Data Processing Segment of the ECS Project*.

The VDD will contain the prioritized current status summary of any Trouble Tickets/Discrepancy Reports against the CI that is being issued per the change request.

Some general guidelines and/or items that must be tailored for the specific size and scope of configuration audit to be conducted include:

(a) Audit Plan;

(b) Conference Agenda;

(c) Location to collect and analyze data; conduct meetings;

(d) Applicable specifications, drawings, manuals, schedules, design and test data;

(e) Test Results Analysis;

(f) Meeting minutes including resulting audit action items;

(g) Tools and inspection equipment necessary for evaluation and verification;

(h) Unencumbered access to the areas and facilities of incoming inspection, fabrication, production, and testing;

(i) Personnel from each engineering, production, and quality department to be available for discussion of their respective areas;

(j) Copies of inspection reports, process sheets, data sheets, and other documentation deemed necessary by the Government FCA/ PCA teams; and

(k) Isolation of the item(s) and detailed parts to be reviewed.

9.5 Data Management

9.5.1 Purpose

The term "data management" pertains to two activities. The first pertains to the activities of the Data Management Organization (DMO) within the ECS project. The second, broader in scope, includes all organizing and use of documents and data across the ECS project. This procedure describes the DMO activities fully. In addition, certain activities that are material to the data management of the project are described in this procedure although they are performed by organizations other than the DMO.

This procedure describes:

- the policies and procedures that apply to data handling and document distribution.
- requirements for data management for the ECS project.
- the role of the data base administrator .
- the use of data control and data handling systems (Document Data Handling Subsystem (Section 20).
- the activities pertaining to establishing and maintaining system libraries and records.
- management of data required during operations readiness review, missions operations, and certification and to assist the on-going development of the system for design, implementation, and test.
- how data will be collected, maintained, and made available to the development team and for distribution to the NASA.
- the data management functions of controlling document masters, preparing change pages, and keeping auditable change records.

- the plan for controlling the data base structure, controlling the interfaces to the data base, establishing the data base security, and evaluating data base performance.

This procedure does not cover the handling or use of notes, test data, software, financial information, or draft documents that would generally be characterized as working papers or other non-controlled and non-deliverable information. This information is handled by the contractor Integrated Management Information System (IMIS).

9.5.2 Applicable to

All ECS Sites CM Administrators and the SEO Librarian

9.5.3 References

Data Management Plan for the ECS Project	104-CD-001-004
Documentation Management and Control	DM-1-001
Data Identification Numbering	DM-1-002
CDRL Item/Required Document Generation, Review, Release & Maintenance	DM-1-004
CDRL Document Format	DM-1-006
Document Delivery and Dissemination	DM-1-007
Documentation Archiving and Storage	DM-1-009
Technical Paper and White Paper Generation, Review, Release, and Maintenance	DM-1-013
Distribution of ECS Documentation and Information Using World Wide Web Technology	DM-1-014
Data Management Handbook	DM-1-017

9.5.4 Procedures

The following text describes standard data management procedures and methods to be used by the ECS project. Occasionally special circumstances may arise which call for exceptions and flexibility to the customary procedures.

9.5.4.1 Information Preparation, Submittal, & Cataloguing

9.5.4.1.1 Creation/Preparation

The originator / author (usually engineering) will create all source material (text, graphics files, etc.) per CDRL/DID preparation instructions and be accountable for the accuracy of its content. Publishing will assist the author by providing Word Processing and Graphics support such as templates and fonts. (Publishing will do the final formatting later.) The DMO will provide the appropriate CDRL numbering and DID instructions.

9.5.4.1.2 Submission

The originator/author will submit all source material (text, graphics files, etc.) to the DMO electronically including necessary metadata descriptors. The latter include reference to source documents and dates.

The Data Management Office verifies delivery schedule with the appropriate task manager prior to a scheduled release or CDRL delivery date. The DMO notifies the responsible organization at the Program Office Weekly Status Report of upcoming CDRL items and their delivery dates, any special preparation instructions, formats, title pages, signature approvals, and any other pertinent information. Copies of the ECS Master Schedule are provided at the Weekly Status Review.

9.5.4.1.3 Identification and Numbering

Data Identification. Upon receipt of information, the SEO Librarian assigns a DMO identification number to each pertinent type of program data created for the ECS program. If appropriate both SEO and NASA numbering are assigned at this time.

9.5.4.1.4 Logging/Cataloguing

The DMO verifies proper submission of information into the system including valid numbering, manually making changes if necessary, and provide supplementary cross-references as part of the data base catalogue. DMO also updates the status log for those submissions that were pre-scheduled.

9.5.4.2 Information Review, Signoff, Release, and Change/Revision

9.5.4.2.1 Document/Test Data Review, Release, and Change Procedures

CDRL Item/Required Document Generation, Review, Release & Maintenance (DM-1-004)

9.5.4.2.2 Review/Release

Prior to its release, each CDRL document will be reviewed and signed off by each of the participating functions involved (e.g., engineering, product assurance, systems engineering) as depicted in Figure 9.5.4.2-1. Only when all the proper signatures have been obtained will the document be released and distributed.

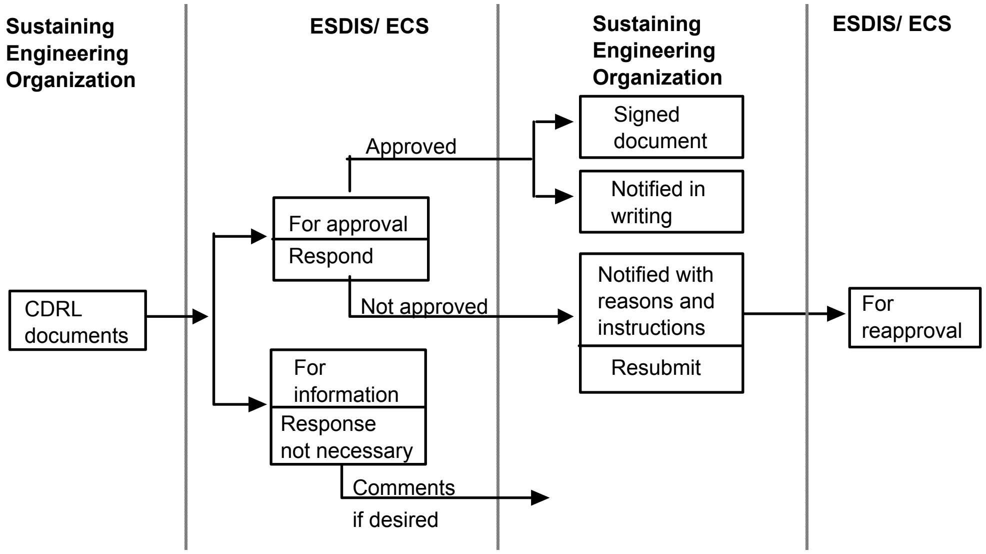


Figure 9.5.4.2-1. CDRL Item Submission Criteria Flow

The DMO will verify that review and signoff has been completed and annotate the status log for the respective information accordingly.

Data Control Procedures:

1. Data Item Scope, Content, and Format
2. Data Item Submittal (Schedule and Distribution)
3. Data Item Use of Common Information
4. Data Not Duplicated
5. Data Item Quality Control and Assurance
6. Data Item Approval
7. Internal Data
8. Data Items Rework
9. Recorded Information

9.5.4.2.3 Changes, Revision, and Document Maintenance

Changes to controlled documentation come under control of the appropriate Change Control Board (CCB) and details concerning the procedure are to be found in the Configuration Management Plan. In summary, once the author (usually engineering) and the CCB agree on a change, the DMO is mandated to reflect the change in the data and documents. Changes to other data come under control of the appropriate task manager.

Once information is released, no changes will be permitted except upon the instruction of the appropriate authority (CCB or task manager). Changes to specifications are submitted to a formal change review board, which repeats the review/approval process (described above).

The DMO receives the document with approval signatures and reviews for compliance with the SOW, DID, and other contractual requirements. If found to be deficient, the document is returned to the author for corrections.

Revision and resubmission of any CDRL item will be subject to the same submission criteria as applied to the initial release of the document. The DMO will store a file copy and include information about updates in the next status report. The DMO will maintain a detailed data catalog and provide, as requested, specific indices that include document revisions and issue dates.

9.5.4.3 Information Distribution and Submission to ESDIS/ECS

9.5.4.3.1 Data/Document Distribution/Submittal to ESDIS/ECS

ESDIS/ECS will receive preliminary versions of data for comments prior to final release. After data in final form is received and cataloged, the DMO will reproduce the data, if necessary, and make distribution in accordance with CDRL/DID instructions and 500-TIP-2601 delivery interchange standard. The DMO will maintain special distribution lists (sorts) for any requirement that may occur. For documents transferred electronically, the Document Data Server Subsystem will encode and transmit the documents to ESDIS/ECS and will prepare and deliver a backup disk or tape for each document in the appropriate format.

All contractual data submitted to ESDIS/ECS, both engineering and non-engineering, will have a standard transmittal sheet attached. This sheet will contain key information about the data being submitted, such as data number, description, submission criteria, format.

9.5.4.3.2 Categories of CDRL Data Submitted to ESDIS/ ECS

All CDRL data submitted to ESDIS/ECS will be classified as being *for information* or *for approval.*

1) For Information. – Routine documentation which will be evaluated by ESDIS/ECS to determine current program status, progress, and future planning requirements. Examples of *for information* documents include status reports and programs management directives. *For information* documents shall be sent to ESDIS/ECS as soon as approved and issued by the SEO. ESDIS/ECS may elect to provide comments, although a formal ESDIS/ECS response is not required.

2) For Approval. – Documentation that requires written approval from ESDIS/ECS before its acceptance, distribution, and intended use. Examples of *for approval* documents include all documentation that is required to come before a CCB. ESDIS/ECS will approve the document or ask for resubmission at ESDIS/ECS Program Office. Provisions will be made for ESDIS/ECS signature on the cover of documents submitted for approval. If the document is approved, ESDIS/ECS will sign and return the document and notify SEO in writing of the approval. If the document is not approved, within a mutually agreed time, ESDIS/ECS will notify SEO of those parts of the document which cannot be approved, together with the reasons and instructions concerning resubmission of the document. If the ESDIS/ECS evaluation reveals inadequacies, ESDIS/ECS will inform SEO of the parts of the document that require alterations, recommend actions, and give resubmission instructions according to the character of the inadequacies. SEO will resubmit the document to ESDIS/ECS for approval after receipt of ESDIS/ECS's notification.

9.5.4.3.3 Documentation Distribution

The DMO will prepare approved CDRL(s) transmittal letters. The DMO will reproduce copies of the letter and data package as required by the CDRL, ship it to those on the approved distribution list, and provide internal distribution as appropriate.

9.6 Archiving Procedures for the SW CM Manager (ClearCase)

9.6.1 Purpose

These instructions establish Configuration Management procedures for the backup of ClearCase Version Object Base (VOB) data, Views, and data delivered to the customer.

9.6.2 Applicable to

All ECS CM Administrators, SW Maintenance Engineers, and Sustaining Engineers

9.6.3 References

Configuration Management Plan for the Science Data Processing Segment of the ECS Project	102-CD-003-004
Archiving Procedures for ClearCase	CM-1-014-1
Delivery Package Archiving	CM-1-034-1

9.6.4 Procedures

DEFINITION

ECS Development Facility (EDF) - the software development environment including data, hardware, software, networks, facilities, and procedures used to support ECS software development and testing.

Software - for the purpose of this instruction, software includes all ECS-developed application software, COTS software, build and environmental instructions, and databases used in the execution of these products.

Segment-level - for the purpose of this instruction, segment-level includes all software development undertaken in the EDF by ECS segments from the initiation of software coding through completion of segment integration and test (I&T).

System-level - for the purpose of this instruction, system-level includes all ECS integration and test activities beginning with installation of segment software in the system I&T files.

Software Development File - a repository for a collection of material pertinent to the development or support of software.

Thread - a set of components (software, hardware, and data) and operational procedures that implement a scenario, portion of a scenario, or multiple scenarios.

Build - An assemblage of threads to produce a gradual buildup of system capabilities.

VOB - Version Object Base. Secure, permanent, mountable file system. Repository for storage of version-controlled data.

View - A unique workspace management or “sandbox management” that provides developers with transparent, file-level access to any version of any element through the use of dynamically-evaluated, user-specified version selection rules.

GENERAL

1. System Administrators maintain a backup for all ECS systems. This procedure documents an additional backup procedure for only ClearCase-specific data. This process is in parallel with System Administrator procedures. ClearCase, the ECS level software tracking tool, contains within itself a number of repositories called Version Object Bases (VOBs). These VOBs maintain all versions of all software elements developed for the ECS project. Additionally, there are disk areas known as Views, which contain file(s) that may be newly created or modified but not yet given or returned to ClearCase control.
2. It is critical that these repositories, VOBs, and Views get backed up synchronously on a daily basis, and that the backup tapes be maintained in such a way as to guarantee their restoration to the system in the event of a catastrophic failure. A View may contain files that were checked out of a ClearCase VOB, modified, but not yet checked back in when the failure occurred. This CM archive procedure uses synchronization techniques to ensure that no data is lost. The System Administrator’s procedure does not include these techniques.
3. Current system and network configurations prohibit the installation of the View storage area on the developers’/ maintainers’ personal workstations. As a result, Configuration Management has assumed the responsibility of backing up the View storage areas from the ClearCase server. However, once the developers/maintainers are provided with workstations that permit the installation of a private View storage area, this responsibility will be shifted to the developer/maintainer.

4. All ECS software is delivered electronically and via magnetic media. Copies of the exact magnetic media that were delivered are made and stored similarly to in-house generated backup tapes.
5. Backups of VOBs and Views are accumulative rather than incremental. Therefore, each tape has all data contained on previous archive tapes plus new files created, modified, etc.
6. A tape set consists of 31 archive tapes. The tapes are numbered from 1 to 31, a tape for each day of the month.
7. All software CM personnel will be familiar with the archive procedures.
8. All tape backup and archives are performed on Triton.

INSTRUCTION

1. A Software Configuration Management person will be assigned the responsibility for tape backup and archives. Daily backups are created every day at 03:30 a.m. via a UNIX cron job. Every workday morning the assigned Software CM person will perform tape archive procedures.
2. The daily backup tapes are obtained from and maintained in a fireproof safe which is kept locked. Key CM personnel hold the combination to the safe. The most current tape (the tape corresponding with the day of the month) is removed from the Triton tape drive and is always placed in the slot farthest to the left in the location marked “Current Backup”. The tape corresponding to the day of the month for the next day is inserted into the tape drive.
3. On the first business day of each week the backup tape from the last business day of the previous week is given to System Administrator personnel for transportation to an off-site storage facility. Periodically the tapes stored off-site are returned to CM for reuse.
4. All tapes will be labeled with the type of data contained. In the case of delivered software, this will include, at a minimum, the title and version of the delivery, the tool used to generate the tape, and the date made.

   Example of a routine weekly backup tape label:

   Weekly Clear Case Backup Tape

   VOBs and Views

For week ending: 01/06/94

Tar formatted.

Person Performing Backup:

Example of a Deployed Software Tape:

194-904-DV3-002

Label: PGS TK2 Test Drivers   Date: April 29, 1994

Test_drivers.tar 2gbyte format

Command to copy file to disk: dd if=## of=file.tar
Command to untar tar file: tar xvf file.tar
where ## = your tape device and file.tar = Name you give tar file

5. During weekends and holiday periods the tape for the next workday will be placed in the tape drive on the last workday prior to the weekend/holiday. This tape will be written to on all days prior to the next workday. In this case should the system crash over the weekend/holiday period, there will be a more up-to-date backup. For example, on Friday, Monday's tape would be placed in the drive. This tape will be written to on Saturday, Sunday, and Monday due to the automated cron job. Should the system crash on Saturday after 03:30 a.m., there would be a backup from Saturday available rather than Friday. On Monday this tape would be removed and replaced with the tape for Tuesday.

6. This process does not archive personal view storage areas resident on personal development workstations (see General 3., above). Therefore, when personal development workstations are assigned, files checked out of ClearCase and under modification are not backed up. The System Administrator can back up these workstations. The user can request through the Help Desk that his development workstation be scheduled for System Administration backup. Alternately, personal backup procedures can be initiated.

7. At the time the tar file for software delivery is created, a second delivery tape will be generated in accordance with delivery procedures. This archive copy will be given to System Administrator personnel for transportation to the off-site storage facility. Incremental delivery tapes will be created, documented, and stored in accordance with this procedure. Delivery archive tapes will be stored for the life of this contract.

9.7 SW Transfer and Installation

9.7.1 Purpose

This procedure involves transferring a Sustaining Engineer Organization (SEO)-developed software maintenance change package from the SMC to a remote site (a DAAC) and later installing the ECS custom software on a selected host computer under a configuration management controlled process. The procedure begins when the SMC Configuration

Management Administrator receives the software maintenance change from the SEO and directs transfer to a designated DAAC drop-off point (SEO on-site SW library). At the DAAC, the installation actions are executed by the site sustaining engineering SW Maintainer under direction from the DAAC Configuration Change Board (CCB).

9.7.2 Applicable to

All ECS sites' Sustaining Engineers, System Administrators, CM Administrators, and Maintenance Engineers.

9.7.3 References

Configuration Management Plan for the Science Data Processing Segment of the ECS Project	102-CD-003-004
Developed Software Maintenance Plan for the ECS Project	614-CD-001-003

9.7.4 Procedures

9.7.4.1 Overview

This procedure details the transfer of SW changes under CM control from the ESDIS CCB using SMC resources to distribute maintenance changes to the sites (nominally, to a DAAC).

Assumptions:

- The SMC storage software will be ClearCase.
- The baseline records will be maintained in the Baseline Manager (ClearCase tool)
- The SMC transfer of software will be via tar tape or File Transfer Protocol (FTP)
- The transfer-storage point will be the SEO on-site SW library.
- The Software Maintenance Change package is relatively small and requires no special build/test procedures.
- Resource Planning, Mode Management, and other issues are not addressed in this scenario.

Summary of Procedures:

- CM Process defined Changes to be incorporated by SEO into ECS Operational Baseline
- SW received at SMC from SEO CM Administrator
- Baseline changed via Baseline Manager (ClearCase tool)
- Packaged via ClearCase
- Transferred via tar tape or File Transfer Protocol (FTP)

- DAAC CCB Approves the Installation of SW Change Package into DAAC Operational Baseline
- SW Change Package Installed at DAAC on selected host computer

9.7.4.2 Operator Roles

SEO CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Maintains control of all configured hardware and software.

SMC CM Administrator--Provides ECS system-wide configuration management and exercise control and/or monitoring over the configurations.

DAAC CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Maintains control of all configured hardware and software. Assists in the development and administration of the library with respect to configuration management procedures.

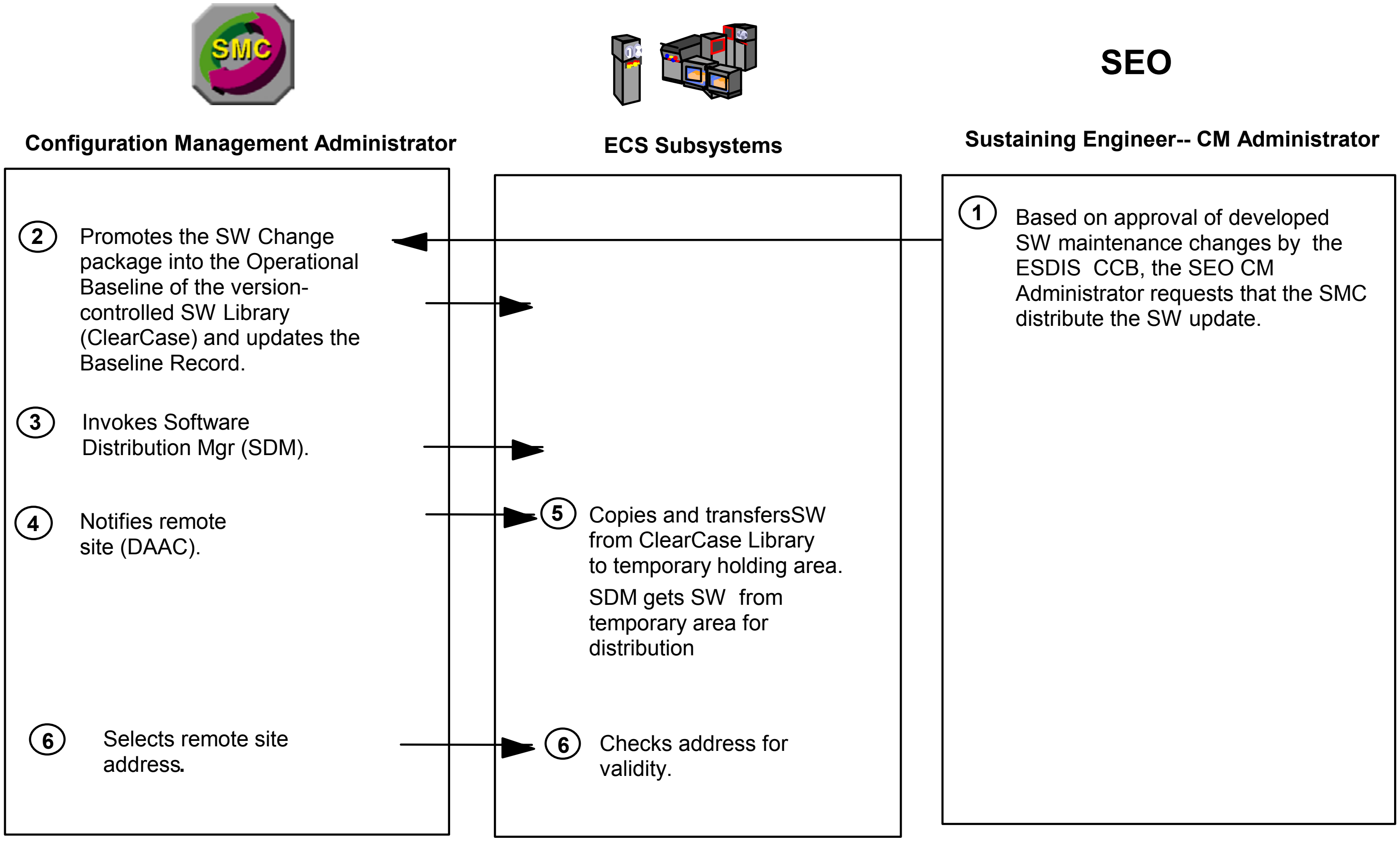
DAAC Sustaining Engineering--SW Maintainer--Produce, deliver, and document the corrections, modifications, and enhancements made to ECS software (including COTS), and/or adapt or incorporate COTS software for ECS use.

9.7.4.3 Detailed Procedures

The following figures are a three part Point of View chart that steps through all the procedure and showing how all relevant roles interact.

Software Transfer & Install

Points of View I (Transfer)



Configuration Management Administrator

2 Promotes the SW Change package into the Operational Baseline of the version-controlled SW Library (ClearCase) and updates the Baseline Record.

3 Invokes Software Distribution Mgr (SDM).

4 Notifies remote site (DAAC).

6 Selects remote site address.

ECS Subsystems

5 Copies and transfersSW from ClearCase Library to temporary holding area.

SDM gets SW from temporary area for distribution

6 Checks address for validity.

SEO

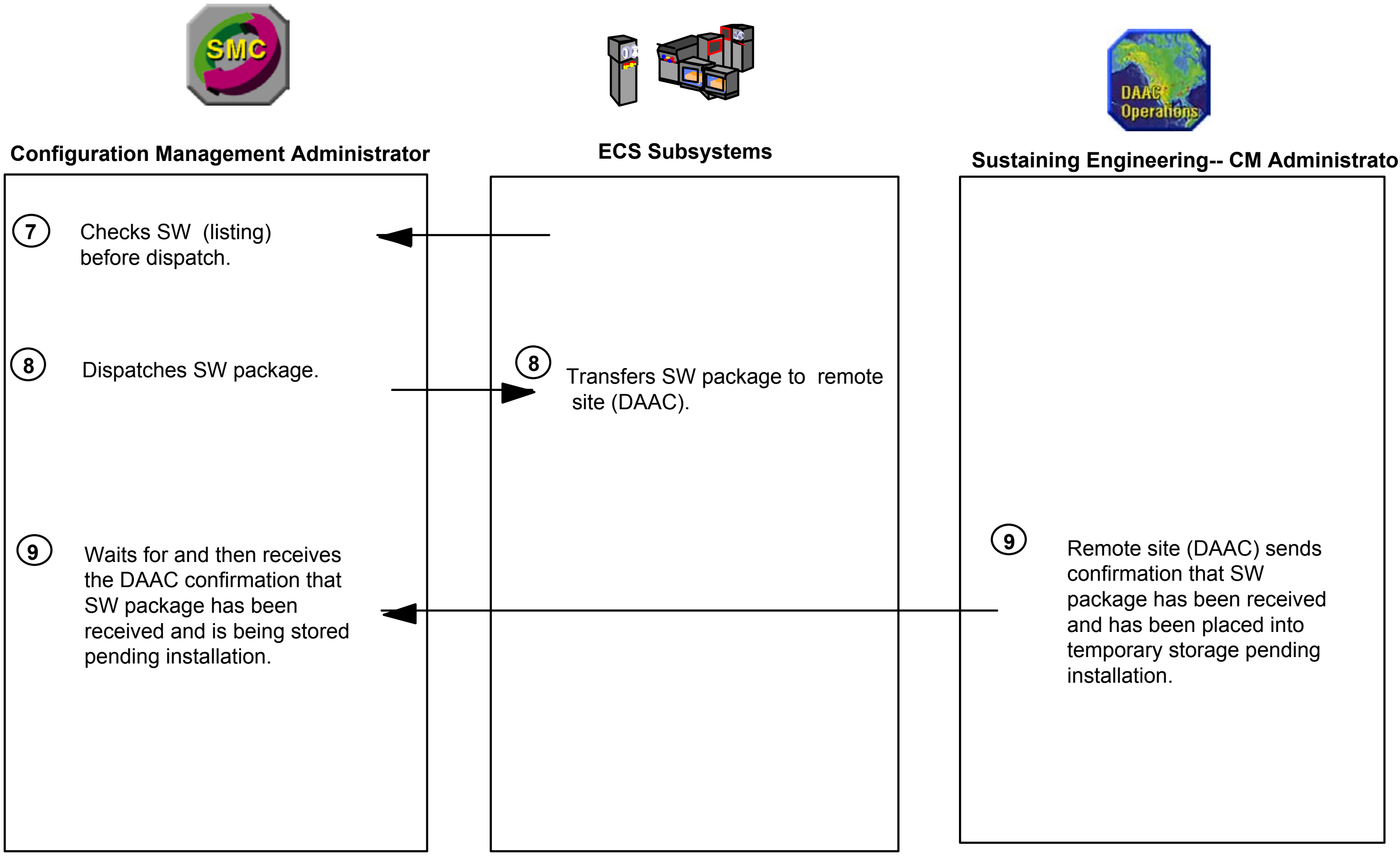
Sustaining Engineer-- CM Administrator

1 Based on approval of developed SW maintenance changes by the ESDIS CCB, the SEO CM Administrator requests that the SMC distribute the SW update.

Figure 9.7.4.3-1. Detailed Points of View

Software Transfer & Install

Points of View II (Transfer)



Configuration Management Administrator

7 Checks SW (listing) before dispatch.

8 Dispatches SW package.

9 Waits for and then receives the DAAC confirmation that SW package has been received and is being stored pending installation.

ECS Subsystems

8 Transfers SW package to remote site (DAAC).

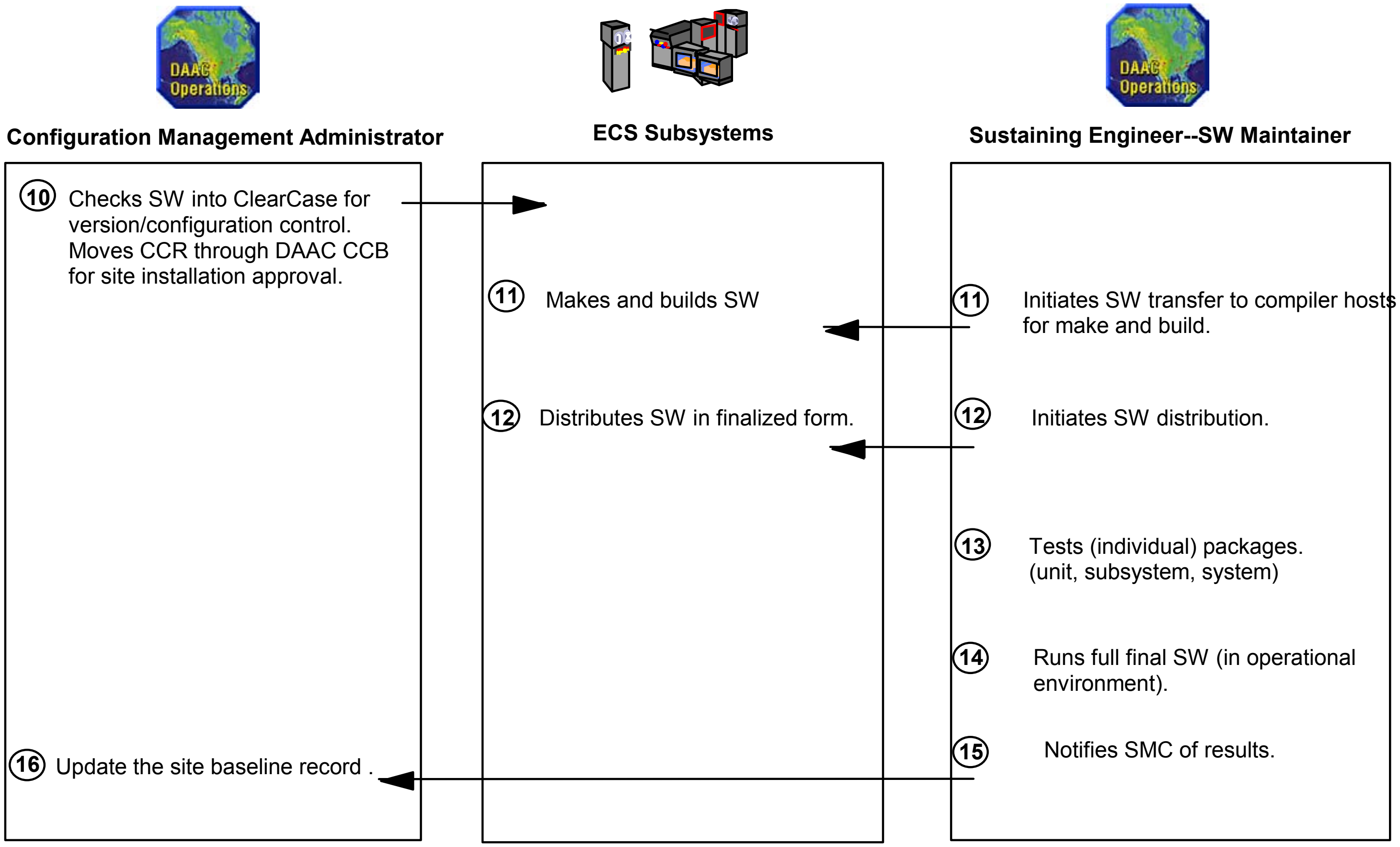
Sustaining Engineering-- CM Administrator

9 Remote site (DAAC) sends confirmation that SW package has been received and has been placed into temporary storage pending installation.

Figure 9.7.4.3-2. Detailed Points of View

Software Transfer & Install

Points of View III (Install)



Configuration Management Administrator

(10) Checks SW into ClearCase for version/configuration control. Moves CCR through DAAC CCB for site installation approval.

(16) Update the site baseline record .

ECS Subsystems

(11) Makes and builds SW

(12) Distributes SW in finalized form.

Sustaining Engineer--SW Maintainer

(11) Initiates SW transfer to compiler hosts for make and build.

(12) Initiates SW distribution.

(13) Tests (individual) packages. (unit, subsystem, system)

(14) Runs full final SW (in operational environment).

(15) Notifies SMC of results.

Figure 9.7.4.3-3. Detailed Points of View

9.7.4.4 Data Activity

The following four tables, 9.7.4.4-1 through 9.7.4.4-4, describe data activities associated with the SW transfer and installation workflows of Figures 9.7.4.3-1 and 9.7.4.3-2 for SMC Configuration Management Administrator's data maintenance activities and 9.7.4 B&C for site-level (DAAC) configuration management data maintenance by the SEO Maintenance Engineer and CM Administrator.

Table 9.7.4.4-1. Data Activity for Workflow at the SMC

SMC Configuration Management Administrator Role	Data Element	Operator Interactions (Edit, Input, Display)
1. Invoke Software	Package ID Package Name SW Upgrade Name Version Description File Structure Type	Input/Edit Input/Edit Input/Edit Edit Edit Edit Input/Edit
2. Select remote site address (DAAC).	Destination	Input/Edit
3. Check, dispatch S/W package.	Destination	Edit

Table 9.7.4.4-2. Data Activity for Workflow at the DAAC

DAAC CM Administrator	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
1. Receives message of SW ready for site installation	E-Mail (send/receive)	Display
2. Processes CCR for DAAC Installation of Software Upgrade install script stored in ClearCase.	DDTS: DAAC CCR # SW Package ID Package Name SW Upgrade Name Version Description File Structure Type Installation Schedule	Input/Edit

Table 9.7.4.4-3. Data Activity for Workflow at the DAAC

DAAC Sustaining Engineer SW Maintainer	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
3. Installs SW Upgrade installation script stored in ClearCase	Package ID Package Name SW Upgrade Name Version Description File Structure Type	Input/Edit
4. Verifies that all of the paths and directory structures have been created and are correct.	ClearCase: Description File Structure Type	Input/Edit
5. Runs all diagnostic tests to verify that the new upgrade is operating as expected.	ClearCase: Description File Structure Type	Input/Edit
6. Informs the Resource Manager that the upgrade is completed.	E-Mail (send/receive)	Input/Edit

Table 9.7.4.4-4. Data Activity for Workflow at the DAAC

DAAC CM Administrator	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
7. Check the Baseline Record	ECS Baseline Information System (EBIS): DAAC CCR # SW Package ID Package Name SW Upgrade Name Version File Structure Type Installation Date	Display (Print)

9.8 Change Request Manager

The commercial off-the-shelf (COTS) product, Distributed Defect Tracking System (DDTS), serves as the Change Request Manager (CRM). DDTS provides the functionality necessary to compose, submit, report, and track status of changes proposed for ECS resources. It provides the M&O staff at the sites and the SMC the capability to register Configuration Change Requests (CCRs). DDTS prompts for relevant information, assigns an identifier, and mails notification of the newly submitted requests to pre-designated personnel. As the CCRs advance through approval and implementation processes, DDTS maintains status, disposition, resolution, and closure information as entered by the M&O staff. It sends notification to pre-designated personnel when the status of the CCR record changes and makes data available for viewing by authorized staff members.

The Activity Checklist table that follows provides an overview of Change Request Manager capabilities. Although Column one (**Order**) shows the order in which tasks might be accomplished, please note that they are independent of each other and can be performed in any order, as necessary. Column two (**Role**) lists the site/organization Configuration Management Administrator (CMA) responsible for performing the task. Column three (**Task**) provides a brief explanation of the task. Column four (**Section)** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Change Request Manager - Activity List

Order	Role	Task	When and Why to Use	Section
1	CMA	View CCR	Operator uses this function whenever he/she wants to quickly view the contents of CCRs in the Index.	(P) 9.8.3
2	CMA	Submit CCR	Operator uses this function whenever there is a new CCR to be entered into the DDTS database.	(P) 9.8.4
3	CMA	Change Status of CCR	Operator uses this function whenever the activities of a particular state have been completed and its time to move to the next state.	(P) 9.8.5
4	CMA	Modify CCR	Operator uses this function to change previously entered data and/or to enter data into fields previously left blank.	(P) 9.8.6
5	CMA	Print CCR	Operator uses this function to obtain a hard or soft copy of a CCR or all of the CCRs in the CCR index.	(P) 9.8.7
6	CMA	Generate Reports	Preformatted reports will be generated for each CCB.	(I) 9.8.9

NOTE: All changes made to the CCR record are monitored by the system and logged in the History enclosure. To view this log, click on the History icon on the DDTS main screen (PureDDTS).

9.8.1 The Configuration Change Request (CCR)

The CCR form has been developed as a medium for processing CCRs throughout the ECS Maintenance and Operations environment for changes processed by the ESDIS CCB and its ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL). The information included on the CCR is described below. Copies of the CCR, CCR Impact Analysis Form, and CCR Impact Summary form are provided in Section 9.2. Each CCB will have unique CCR identification sequence numbers. Each CCB can forward CCRs and reports from the Change Request Manager to SMC, where SEO processes system-level CCRs for ESDIS CCB. The ESDIS CM Plan will determine the charter of the respective CCBs and thus the scope of CCR issues to be addressed by the site CCBs.

Section 9.8.4 defines the procedures to enter data from the hardcopy CCR into the Change Request Manager.

Many of the numbered items on the form correspond to the data entry required for CCRs maintained in the Change Request Manager. [Where the hardcopy CCR information is entered in the Change Request Manager tool is defined by referencing appropriate tables found in the subsections that follow.] The Configuration Management Administrator oversees maintenance of the CCR records in the Change Request Manager for his or her respective CCB.

1. **Configuration Control Board (CCB)** — The designated CCB is checked-off for changes processed by the ESDIS CCB and its ECS site-level chartered CCBs at the SMC, DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL), and EOC. [This information is not entered into the Change Request Manager.]

2. **CCR Number** — The unique serialized CCR number is applied at each site. This number is system-generated. [Submit record field; see Table 9.8.4-1.]

3. **Submitted Date —** The date that the CCR was prepared is documented. [Submit record field; see Table 9.8.4-1.]

4. **Revision** — The current revision is designated for tracking changed versions of the original CCR. The Revision number is assigned by the Configuration Management Administrator or by the originator of the CCR with approval of the Configuration Management Administrator. [Submit record field; see Table 9.8.4-1.]

5. **Priority —** The priority level of the CCR is assigned by the CCB. Emergency CCRs may have already been implemented on a temporary basis by the Trouble Ticket Review Board (TTRB) with concurrence from the CCB Chair who later receives the CCR to document/implement the permanent change. Urgent items will be reviewed by the next CCB meeting. Routine items will be reviewed as soon as the schedule permits. [Submit record field; see Table 9.8.4-1.]

6. **Change Class** — Change Classes are either I or II. Class I will be handled by ESDIS-only because of cost, schedule, and/or mission impacts that may require requirements changes. Class II items do not affect mission requirements but may have cost and/or schedule implications which affect maintenance, operations, procedures, documentation, site-tailored items, COTS implementation, site installations of core system changes, science SW changes, etc. [Submit record field; see Table 9.8.4-1.]

7. **Status —** Status of the CCR is updated in the Change Request Manager until closed by the CCB. Note that the hard copy form will not be updated but will be kept in the master suspense file of the CM Administrator until closed-out with a stamp (Item 15, below) and appropriate signatures (see Items 16 and 17, below). [Submit record field; see Table 9.8.4-1.]

   Eleven valid status indicators (states) are listed below. The corresponding Change Request Manager State Code (upper-case single character) is provided in parentheses after the status descriptive name. This code is stored in the change request record. The Change Request Manager uses this value to search and extract the descriptive name to display in reports. The descriptive names, for example, Assign-Eval, are as they appear in the Change Request Manager for selection during input. Some query and report codes use the descriptive name rather than the single letter code to facilitate querying. Following the definition is the status as it appears on the CRR. These status indicators appear in the Change_State Menu, described in Section 9.8.5, below.

Submitted (S) — not used on CCR hardcopy but is system-generated when CCR is input to the Change Request Manager prior to a CCB decision. (Submitted)

New (N) — the initial state for all newly entered change requests. (New)

Assign-Eval (A) — state entered when the change request is being assigned to an engineer for evaluation/analysis. (Assigned-Eval)

Assign-Implement (O) — state entered when the change request is being assigned to an engineer for development. (Assigned-Implement)

Implement (R) — state entered when the proposed change has been developed. (Implemented)

Assign-Verify (T) — state entered when the developed change is being assigned to an engineer for verification testing. (Assigned-Verify)

Verify (V) — state entered when a developed change has been tested and verified that it functions properly. (Verified)

Close (C) — state entered when all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity. Referred to as Close-out stamp. (Closed)

Duplicate (D) — state entered when a change request is determined to be a duplicate of an existing change request. Duplicate change request identifies change request being duplicated. (Duplicate)

Defer (P) — state entered when activity on a change proposal has to be postponed. (Deferred)

Forward (F) — state entered when a change request needs to be forwarded to another DDTS-defined project. In DDTS terminology, a project is a grouping of change requests. For example, a change request from a site project can be forwarded to an ECS project. (Forwarded)

8. **CCR Title** — The CCR title is supplied by the originator. [Submit record field; see Table 9.8.4-1.]

9. **Originator** — The originator name, organization, e-mail address, and phone number is given. [Submit record fields; e-mail address not included; see Table 9.8.4-1.]

10. **Approval** — The CCR is approved by the designated management authority which is assigned by the CCB. This sponsorship requirement acts as a primary filter to eliminate from consideration those CCRs that cannot be implemented or which have no ECS site management support. [Assign-Implement field; see Table 9.8.5-2.]

11. **Reason for Change** — The reason for the change is narrated on the form and/or the designated attachment. [This information may be included in the Proposed Change Enclosure; see Section 9.8.4-1, Step 2.]

12. **Description of Change** — The proposed implementation of the change is narrated along with any known impacts, resources, and expenses to be incurred. [This information may be included in the Impact Summary Enclosure; see Section 9.8.5.1, Step 2.]

13. **Impact Analysis** — Impact analysis is documented in the CCR Impact Analysis form. The impact analysis is collected by the CCB Chair appointed Evaluation Engineer in coordination with the CM Administrator who maintains the CM records and assembles the review package for the CCB. The Evaluation Engineer documents the list of Impact Evaluators and derives and/ or verifies cost, technical, and schedule impact of the proposed change based on all inputs received. The results of the coordinated CCR Impact Analysis inputs are presented in the CCR Impact Summary form as part of the CCR review package. [This information may be included in the Impact Summary Enclosure; see Section 9.8.5.1, Step 2.]

14. **Comments** — Comments are added to the CCR to summarize sites and/or organizations affected by the CCR. Additional comments may address proposed CCB dispositions and recommendations to be indicated by resolutions in Item 15, below. [This information may be included in the Resolution Enclosure; see Section 9.8.5.2, Step 2.]

15. **Board Action** — CCB actions and follow-up actions that will be facilitated and tracked by the CM Administrator are indicated. [Assign-Implement field; see Table 9.8.5-2. Also may be applicable to Resolution Enclosure; see Section 9.8.5.2, Step 2.] Possible CCB dispositions are given as approved, withdrawn, disapproved, and deferred (pending follow-up activities by the indicated schedule date). Further actions are indicated as:

    Engineering Change Proposal (ECP) — changed scope of contract requirements.

    Waiver — declaration that certain contract requirements no longer apply.

    Deviation — change of contract terms or substitution of terms or deliverable requirements.

    Technical Direction — order by Contracting Officer's Technical Representative (COTR) to perform certain tasks within the scope of the contract.

    Contract Modification — changes to the terms of a contract.

    Document Change Notice (DCN) — notification of changes to published documents.

    Others — Engineering Change Notice, Change Order, Escalation to higher CCB authority, etc.

16. **CCB Approval** — CCB approval signature authority by CCB Chair or designate. [Assign-Implement field; see Table 9.8.5-2.]

17. **CCR Implemented** — This signature and close-out stamp (Item 7, above) are executed by the CM Administrator witnessing the completion of the CCR implementation process,

which is tracked in the Change Request Manager automated tool and updated in Baseline Manager for affected version control status changes. [Assign-Implement field; see Table 9.8.5-2.]

Sections 9.8.3 through 9.8.7 define procedures to process a CCR using the Distributed Defect Tracking System (DDTS) application software database tool that implements the Change Request Manager. The procedures, though step-by-step, are not detailed for the novice user. Please refer to the PureDDTS User's Manual whenever further explanation may be required. Relevant sections of the Manual are identified where applicable.

9.8.2 Accessing Change Request Manager

Depending on your site configuration, access to the Change Request Manager, DDTS, will be by clicking an icon from your desktop, or by typing the following at the command line prompt:

➢ **xddts**

The PureDDTS screen is the main screen. It consists of three major areas:

- the CCR Index Display which shows a listing of CCRs;
- the CCR Record page, which displays some of the content of the highlighted CCR in the Index; and
- the Enclosure Display, which shows the initial set of enclosure icons available for CCR update.

From this screen, you initiate all DDTS functions: View CCR, Submit CCR, Change state of CCR, Modify CCR, Print CCR. Reference Chapter 3 of the PureDDTS User's Manual for information concerning the menus and buttons on the DDTS Main Screen.

9.8.3 View a CCR

Entering DDTS brings you to the main screen (PureDDTS). To view any CCR listed in the CCR Index, simply highlight the desired CCR. The CCR is accessed.

9.8.4 Submit a CCR

Clicking the Submit button on the main screen (PureDDTS) will bring up the "Submit A New Change Request" screen. This screen enables the operator to select a class of projects (the Change Request Class is the default class) and a specific project (group of CCRs within the selected class) to which he/she wants to add a CCR. Reference Chapter 2 of the PureDDTS User's Manual for a detailed explanation of the terms, class and project.

As stated above, the selection for class of projects is defaulted for CCR processing, so you won't need to change it:

Submit to which class of projects: Change_Request

To select Project name, either type in your selection or type a question mark as shown:

Project name: ?

A drop-down menu will appear from which to make your selection. (The Configuration Management Administrator can add a project to the list. See the PureDDTS Administrator Manual.)

Click on **Help** to get an explanation for each of the fields shown, how to move within a screen, and how to terminate the submit process.

Once the Configuration Management Administrator enters the desired class of projects and project name, the CCR page displays the CCR record form. This form enables the operator to enter detailed information concerning the proposed change request. Descriptions of the Submit Record fields are listed in Table 9.8.4-1. Table 9.8.4-2 presents the steps performed using DDTS to submit CCRs. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided in this section.

Table 9.8.4-1. Submit Record Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
CCR Number	character	10	System generated	a unique identifier for this resource change request
Submitted	date	6	System generated	the date this proposed change was first registered
Revision	character	2	Optional	the current revision/amendment to the proposed change
Priority	character	9	Required	the urgency with which a proposed change is needed. Answer must be one of the following: routine, urgent, emergency. The default is routine
Change Class	character	2	Required	the classification that distinguishes change requests according to management level needed for approval. Answer must be I or II. The default is II.
Status	character	17	System generated	the stage this proposed change has reached in its life cycle
Title	character	72	Required	the nomenclature used to identify the proposed change

Table 9.8.4-1. Submit Record Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Originator Name	character	25	Required	name of the person who is the author of the proposed change
Organization	character	30	Required	name of the originator's organization
Phone Number	character	13	Required	phone number where originator can be reached
Organization Evaluation Engineer	character	25	Required	name of the person who initially determines whether or not the proposal has merit and should be entered into the DDTS database
CM Admin. Name	character	8	System generated	name of the individual who registered this proposed change/enters the proposed change into the DDTS database. Note, DDTS uses User's Login ID
Organization	character	5	Required	name of the CM administrator's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC
Phone Number	character	13	Optional	phone number of the CM Administrator

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. Enter data in the Submit Record fields.
3. Display the Proposed Change Enclosure Screen by traversing all of the CCR record fields or by clicking on the Proposed Change icon. This enclosure is used to hold additional information about a proposed change. It enables the operator to enter a free text description of the perceived need or problem and a proposed solution. For more information on the enclosure screen see Chapter 3 (Enclosures Section) of the PureDDTS User's Manual.
4. Click the File menu on the enclosure screen and select its “save as” option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).
5. When the main screen display reappears, click the “Commit” button to store the CCR record into the DDTS database.

Table 9.8.4-2. Submit a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Submit button.	Initiate the CCR record submission process.
2	Enter data in fields and Proposed Change enclosure.	Enter data in Record screen.
3	Click on Commit button.	Store record into DDTS database.

9.8.5 Change State of CCR

The first status (state) assigned to a CCR after it is committed to the DDTS database is "New." Refer to the upper left corner of the center section of screen for the current status of the CCR. When it is time to move the CCR to its next life cycle state, the Change_State Menu on the main screen is used. See Number 7, Status, in Section 9.8.1, above. Table 9.8.5 presents the steps performed using DDTS to change the state of a CCR. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below and to Sections 9.8.5.1 through 9.8.5.5.

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. Click on the Change_State Menu to access the available state options available for the CCR record.
3. Select the next state to be assigned. After the next state is selected, the associated data fields (if any) for this new state are accessed.
4. Enter data into the associated data fields for the state, as indicated on the screen. These vary according to state being entered. For descriptions of these states, see Sections 9.8.5.1 through 9.8.5.5: Assign-Eval, Assign-Implement, Assign-Verify, Verify, Close. For the states Duplicate, Defer, and Forward, refer to the DDTS User's Manual (also Section 9.8.1, above).

Table 9.8.5-1. Change State of a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Change_State menu.	Select State.
2	Enter data in fields	Enter data in applicable record screen.
3	Click on applicable Enclosure icon.	Enter data.
4	Click on Commit button.	Store record into DDTS database.

9.8.5.1 Assign-Eval State

The Assign-Eval state indicates that the change request is being assigned to an engineer for evaluation/analysis. Table 9.8.5.1-1 lists the fields for which values are entered.

Table 9.8.5.1-1 Assign-Eval Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Evaluation Engineer	character	8	Required	name of the responsible engineer designated to analyze the proposed system change. Use Login user name of the engineer
Organization	character	5	Required	name of the evaluation engineer's organization. Answer must be one of the following: SEO, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, SMC, EOS
Eval. Engr. Email Address	character	25	Optional	electronic mail address of the evaluation engineer
Impact Evaluators (evaluators 1 -12)	character	5	Optional	collection of names of organizations designated to assess the impact of this proposed change. Answer (s) must be from the following: SEO, ESDIS, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, SMC, EOC, EDF
Sites Affected (sites 1-9)	character	5	Optional	the collection of names of ECS sites affected by this proposed changes. Answer (s) must be from the following: SMC, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, EOC

Table 9.8.5.1-1. Assign-Eval Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Related CCR#	character	10	Optional	the number of another CCR that is related to/associated with this CCR
CI Affected	character	15	Optional	the identifier of the principal configuration item affected by this proposed system change
Docs. Affected	character	56	Optional	the documents identifiers of the system documents affected by the proposed system change
Release Affected	character	10	Optional	the ECS release in which the proposed change is targeted for implementation
Baselines Affected	character	56	Optional	the identifiers of system baselines affected by the proposed change

1. Enter data in the Assign-Eval fields.
2. Display the Impact Summary Enclosure Screen by traversing all of the Assign-Eval record fields or clicking on the Summary Enclosure icon. This enclosure is used to hold free text information concerning the impact of the proposed change based on inputs received from the evaluators.
3. Click the File menu on the enclosure screen and select its “save as” option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).
4. When the main screen display reappears, click the “Commit” button to store the next state data into the DDTS database.
5. The selected state, “Assigned-Eval,” is now shown as the current state (Status) of the CCR record.

9.8.5.2 Assign-Implement State

The Assign-Implement state indicates when the change request is being assigned to an engineer for development. Table 9.8.5.2-1 lists the fields for which values are entered.

Table 9.8.5.2-1. Assign-Implement Field Descriptions

Field Name	Data Type	Size	Entry	Description
Disposition	character	14	Required	the final decision made by a designated approval official concerning this proposed change. Answer must be one of the following: Approved, Approved_w/cmt, Disapproved, Withdrawn, Deferred
CCB Approval Official	character	25	Required	name of the individual whose decision is reflected in the proposed change's disposition
CCB Approval Date	date	6	Required	the date the final decision was made concerning this proposed change. Required format is yymmdd
CCB Org.	character	5	Required	the name of the organization whose configuration control board have authority to approve the change request. Answer must be one of the following: ESDIS, SMC, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL
Implementation Organization	character	5	Required	name of the organization assigned to implement this proposed change. Answer must be one of the following: SEO, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL
Implement. Engineer	character	8	Required	name of the responsible engineer designated to implement the proposed system change. Use Login user name of the engineer
E-mail Address	character	20	Optional	electronic mail address of the implementing engineer
Start Date-	date	6	Required	date implementation activity is to begin. Required format is yymmdd
Estimated Time to Complete	character	20	Optional	estimated time it will take to develop and unit test proposed change in days or months
Completion-Date	date	6	Optional	the date that the proposed change was completed. Required format is yymmdd
Effective Date-	date	6	Optional	the date that the proposed change is go into operation. Required format is yymmdd

1. Enter data in the Assign-Implement fields.

2. Display the Resolution Enclosure Screen by traversing all of the Assign-Implement record fields or by clicking on the Resolution Enclosure icon. This enclosure is used to hold free text description of the solution for the proposed change request.

3. Click the File menu on the enclosure screen and select its “save as” option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).
4. When the main screen display reappears, click the “Commit” button to store the Assign-Implement state’s data into the DDTS database.
5. The Implement state (state entered when the proposed change has been developed) is the state that follows Assign-Implement on the Change_State Menu. No data fields are associated with the Implement state. When Implement is selected, the status is simply changed to Implemented.

9.8.5.3 Assign-Verify State

The Assign-Verify state indicates that the developed change is being assigned to an engineer for verification testing. Data fields appear under the heading, “TESTING INFORMATION” on the Record screen. Table 9.8.5.3-1 identifies these fields.

Table 9.8.5.3-1. Assign-Verify Field Descriptions

Field Name	Data Type	Size	Entry	Description
Test. Engr. Name	Character	25	Required	Name of the engineer designated to test the system change
Test Org.	character	5	Required	name of the test engineer's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC
Est. Testing Completion Date	date	6	Optional	the date that the tester expects to have completed his testing activity. Required format is yymmdd

1. Enter data in the Assign-Verify fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the “Commit” button to store the Assign-Verify state’s data into the DDTS database.

9.8.5.4 Verify State

The Verify state indicates that a developed change has been tested and verified that it functions properly. The data fields appear under the heading, “VERIFICATION INFORMATION” on the Record screen. Table 9.8.5.4-1. identifies these fields.

Table 9.8.5.4-1. Verify State Field Descriptions

Field Name	Data Type	Size	Entry	Description
Test Status	Character	1	Required	this is an indicator as to whether or not the item (s) being tested has passed the test. Answer must be Passed or Failed
Enclosure Added	Character	1	Required	this is an indicator as to whether or not an enclosure has been to further describe the testing activity. Answer must be Yes or No

1. Enter data in the Verify fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the "Commit" button to store the Verify state's data into the DDTS database.

9.8.5.5 Close State

The Close state indicates that all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity. Data fields appear under the heading, " CLOSING INFORMATION" on the Record screen. Table 9.8.5.5-1 identifies these fields.

Table 9.8.5.5-1. Close State Field Descriptions

Field Name	Data Type	Size	Entry	Description
Closed By	character	25	Required	name of the individual that is closing the CCR
Closing Date	date	6	Required	date that the CCR is closed. Required format is yymmdd
Closer's Organization-	character	5	Required	name of the closing official's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC

1. Enter data in the Close fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the "Commit" button to store the Close state's data into the DDTS database.

9.8.6 Modify CCR

There will be times when the operator needs to change the information that was entered previously into the database or to enter information into fields that were not completed initially. The Modify Menu enables modification of database data. Table 9.8.6-1 presents the steps performed using DDTS to modify a CCR. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below:

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. From the Main Menu (PureDDTS) select the CCR you want to modify.
3. Click the "Modify" menu on the main screen Record to bring up the modify options.
4. Select the "Modify Record" option to change existing information and/or to enter information into fields left blank previously.
5. The cursor appears at the first field that may be modified. The modify record mode enables the operator to go through all of the fields that are associated with the current status of the CCR and make changes where appropriate.
6. Once the changes have been made, return to the main screen (Pure DDTS) and click the Commit" button on the main screen to add the changes to the database.
7. Reference Chapter 3, "Modify Menu and the Enclosure Sections," of the PureDDTS User's Manual for additional information as necessary.

Table 9.8.6-1. Modify a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	From Record screen, click on Modify menu.	Select option.
2	Enter data in fields	Modify record and/or enclosure.
3	Click on Commit button.	Store record into DDTS database.

9.8.7 Print CCR

The DDTS Print option allows the operator to display a CCR or several CCRs in the CCR index in a selected format on his or her monitor; print a CCR or several CCRs in the CCR index to a printer; or print a CCR or several CCRs in the CCR index to a designated file. Table 9.8.7-1 presents the steps performed using DDTS to print CCRs. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below:

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. From the main screen (Pure DDTS), select CCR or CCRs to be printed.
3. Click on the Options Menu then select **Print**, or click on the "Print" button, to bring up the Printing Options Screen. This screen provides the operator the capability to print a highlighted CCR or all of the CCRs in the index on the main screen in a full page, index, one-line, or three-line format.
4. Select the desired option under "Where to Print" and provide appropriate information for printer and file options. Refer to Chapter 3, "Setting PureDDTS Options," of the PureDDTS User's Manual for additional details about the Printing Options screen.

Table 9.8.7-1. Print a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Print button.	Select options.
2	From Format to Print screen, click on Print button..	Store record into DDTS database.

9.8.8 Required Operating Environment

9.8.8.1 Interfaces and Data Types

The DDTS application will interface only with the ClearCase application at each of the sites. This DDTS/ClearCase interface is facilitated by a ClearCase/DDTS Integration COTS package. DDTS has no interfaces external to the ECS.

9.8.8.2 Databases

The PureDDTS database is a proprietary database that supports the SQL 89 standard. Reference Appendixes F and G of the PureDDTS Administrator's Manual for details about the PureDDTS database.

9.8.8.3 Database Schema

Reference Appendix F of the PureDDTS Administrator's Manual for a description of DDTS' database schema.

9.8.8.4 Database Parameters

Reference Appendixes F and G of the PureDDTS Administrator's Manual for DDTS' database parameters.

9.8.8.5 Command Line Interface

DDTS interfaces with ClearCase only. This interface is facilitated through a ClearCase/DDTS Integration COTS package. No custom command line interface has been developed. This interface supports a verification query by ClearCase for authorization by maintenance engineering staff to change configuration controlled files based on specific CCR authorization.

9.8.8.6 Event and Error Messages

Standard DDTS event and error messages are used. There are no messages unique to the ECS implementation. A list of the PureDDTS event and error messages is not provided in the PureDDTS User's and Administrator's Manuals. However, messages provided during execution of DDTS are self explanatory.

9.8.9 Reports

Standard DDTS reports are to be used. Reference Chapter 3 of the DDTS User's Manual (Setting PureDDTS Options) for information concerning the printing of a CCR report and a description of the available report formats.

Table 9.8.9-1. Reports

Report Type	Report Description	When and Why Used
Not Applicable		

9.8.9.1 Sample Reports

9.8.9.2 Sample Report (Full Page Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (full page format).

```
                 ECS_CHNG_REQ
                                    Page 1/3
CCR Number: MSSdd00630  Submitted   : 960529   Revision:
Priority  : routine     Change Class: II
Status    : Closed      Enclosures  : 3
Title:
 Revise Data Input Screen (Example Only)

CCR ORIGINATOR INFORMATION
  Originator Name: Frank Pace
  Organization   : LaRC
  Phone Number   : (999)234-1289
  Organization Evaluation Engineer: J. Bellamy

CONFIGURATION MANAGEMENT ADMINISTRATOR
  CM Admin. Name: bfloyd
  Organization  : LaRC
  Phone Number  : (999)234-1830
                 ECS_CHNG_REQ
                                    Page 2/3
CCR Number: MSSdd00630

ANALYSIS INFORMATION
  Evaluation Engineer : bfloyd
       Organization : LaRC
       Email Address: bfloyd@larc.com

  Impact Evaluators:
       1. GSFC    2. LaRC   3. EDF    4.       5.       6.
       7.        8.        9.       10.     11.      12.

  Sites Affected:
       1. GSFC    2. LaRC    3. SMC    4.       5.      6.
       7.        8.        9.

  Related CCR# :
  CI Affected  : Planning CSCI
  Documents Affected:
  Release Affected  : Release X
  Baselines Affected:
                 ECS_CHNG_REQ
                                    Page 3/3
CCR Number: MSSdd00630

DISPOSITION: Approved              TESTING INFORMATION:
  CCB Approval Official: John Wana     Engr. Name: Joe Tester
  Date: 960607                     Organization: LaRC
  CCB Organization: ESDIS           Est. Testing Completion
```

```
                     Date: 960614

IMPLEMENTATION              VERIFICATION INFORMATION:
  Organization: SEO           Test Status (Pass/Fail): P
  Engineer: bfloyd            Enclosure Added (Y/N): N
  E-mail: efinch@eos.com
  Start Date: 960610
  Est. Time to Complete: 2 days
  Completion Date: 960612
  Effective Date: 960710

CLOSING INFORMATION:
  Closed by: Authur Closer     Date: 960618    Org.: SMC

************ Proposed Change ************
Need or Problem:  Describe the need or problem.

The need is -----

Proposed Solution:  Describe the proposed solution.

Suggest that the following capability be changed as follows:
- capability changes

************ Impact Summary ************
Summarize the impact statements received from the organizations requested to provide impacts.

Summary of impacts received from GSFC and EDF is ---------

Resources Summarized:  [ description of resources ]

Technical Summary:

ROM Summary (BOE, Cost & Schedule):

Recommendation:  [ Insert Recommendation ]
************ Resolution ************
Describe how the request will be resolved/completed.

This request will be resolved as follows:
- Capability x will be modified to ----.

************************************
```

The Resolution Enclosure is the end of the full page format report.

9.8.9.3 Sample Report (Three Line Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option, Three Line Format. [Note: this document's margins forced the data to a fourth line.]

```
Submitted 960529, CCR# MSSdd00630, Originator Frank Pace
Title Revise Data Input Screen (Example Only)
Priority routine, Class II, CCB Org. ESDIS, Disp. Approved, Status Closed.

Submitted 960521, CCR# MSSdd00617, Originator Joseph Winkler
Title Add GUI to X11 Program (Example Only)
Priority routine, Class II, CCB Org. LaRC, Disp. Approved, Status Implemented.
```

9.8.9.4 Sample Report (Index Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (Index format). Fields displayed are CCR Identifier, Title, Change Class, Priority, and Status.

```
MSSdd00630 Revise Data Input Screen(Example Only) II routine C
MSSdd00617 Add GUI to X11 Program (Example Only) II routine R
```

9.8.9.5 Sample Report (One Line Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (One Line format). The operator selects the fields desired for the one line format. In this case, the Identifier, CCR Originator, Originator Organization, Implementing Organization, and Status fields were selected and their data values are displayed.

```
MSSdd00630 Frank Pace        LaRC          SEO   C
MSSdd00617 Joseph Winkler      GSFC          LaRC  R
```

9.8.9.6 Report Customization

Refer to Chapter 8 of the PureDDTS Administrator's Manual for explanations of how to customize DDTS reports. Chapter 8 explains how to customize canned reports and how to create and add new reports.

9.9 Use of the Baseline Manager

9.9.1 Overview

The ECS provides a Baseline Manager (BLM) tool to assist in documenting changes to the baseline, and to maintain a historical record of those changes. The BLM tool is used at the ECS Development Facility (EDF) and/or the System Monitoring and Coordination (SMC) function to maintain system-level records and site-level records; baseline reports are accessible at the operational sites.

Baseline Manager (BLM) is used to record and report the as-built operational baseline of ECS products. It contains the configuration record for each baselined product. It identifies products by CI name, description, location, model/version, and component configured articles. It provides traceability to previous configurations. The BLM tool is an application of *ClearCase*. It provides access to functions for maintaining control item and bill of material information.

9.9.2 Baseline Terms and Concepts

Baseline management is a process to identify and control baselined versions of hardware and software, to provide a standard configuration of systems throughout all sites, and allow unique site-configured systems and baselines. It identifies interdependencies between hardware and software items, and permits maintenance of a complete history of baseline changes throughout the life of the project. For ECS baseline management and BLM tools, certain terms and concepts are key to understanding how data on the system baseline are stored and tracked.

Control Item – any ECS item under version control by Configuration Management.

Configuration Item – an aggregation of hardware, firmware, software, or any discrete component or portion, which satisfies and end user function and is designated for configuration control.

Baseline – a configuration identification document or set of such documents formally designated by the Government at a specific time during the life cycle of a configuration item (CI).

Configured Article – a control item reportable as part of the Configured Articles List (CAL).

CIL – a Configuration Items List (CIL) identifies the approved set of CIs that are subject to CM requirements and procedures.

CAL – a Configured Articles List (CAL) describes all CIs, critical item hardware and software, and supporting documentation by which the exact configuration definition of the hardware and software can be determined.

Additional terms, some of which address specific entries in the BLM tool, further define how data on the system baseline items and structure are tracked.

Assembly – an item made up of other items. A *Parent* item is a higher-level item (e.g., an assembly), which may have one or more *Child* items, or components.

Bill of Material – the list of items that comprise an assembly.

Product Structure – the parent-child pairings that define the bill of material for an assembly; each product structure record specifies the effective dates and quantities for a single component of a parent for each engineering change.

Active Date – the date a component becomes effective in an assembly's bill of material.

Inactive Date – the date a component is no longer effective in an assembly's bill of material.

Engineering Change – a mechanism for grouping, reporting, and controlling product changes collectively

Revision – the sequence number of a product structure change to an assembly; it signifies a change to the configuration of an assembly that does not alter its form, fit, or function.

Implementation Status – a record describing the deployment of a control item to a site and the current state and associated date of its implementation; each control item has one record for each site to which it is deployed.

Exporting Data – creating a formatted file or records extracted from the BLM database; control item engineering change, product structure, and interdependency records may be extracted and sent to another BLM site via ftp.

Importing Data – loading BLM data from a formatted file.

At the lowest level, the baseline is composed of configured articles that are the specific types of items that make up ECS and are tracked using the BLM tool. It is important to recognize, however, that we impose a conceptual structure on those configured articles to help us think about the system. In fact, it is possible to conceptualize the structure of the system in a number of different ways, and we may select a different conceptual structure based on the requirements of the situation. The ECS baseline management approach and the BLM tool permit recording and tracking these different conceptual baselines, which can be related to the same records of the configured articles.

For example, system designers may conceptualize the system in terms that will help them track subsystems and the configuration items for which each subsystem team is responsible. This may produce a baseline structured according to a design view, such as that shown in Figure 9.9.2-1.

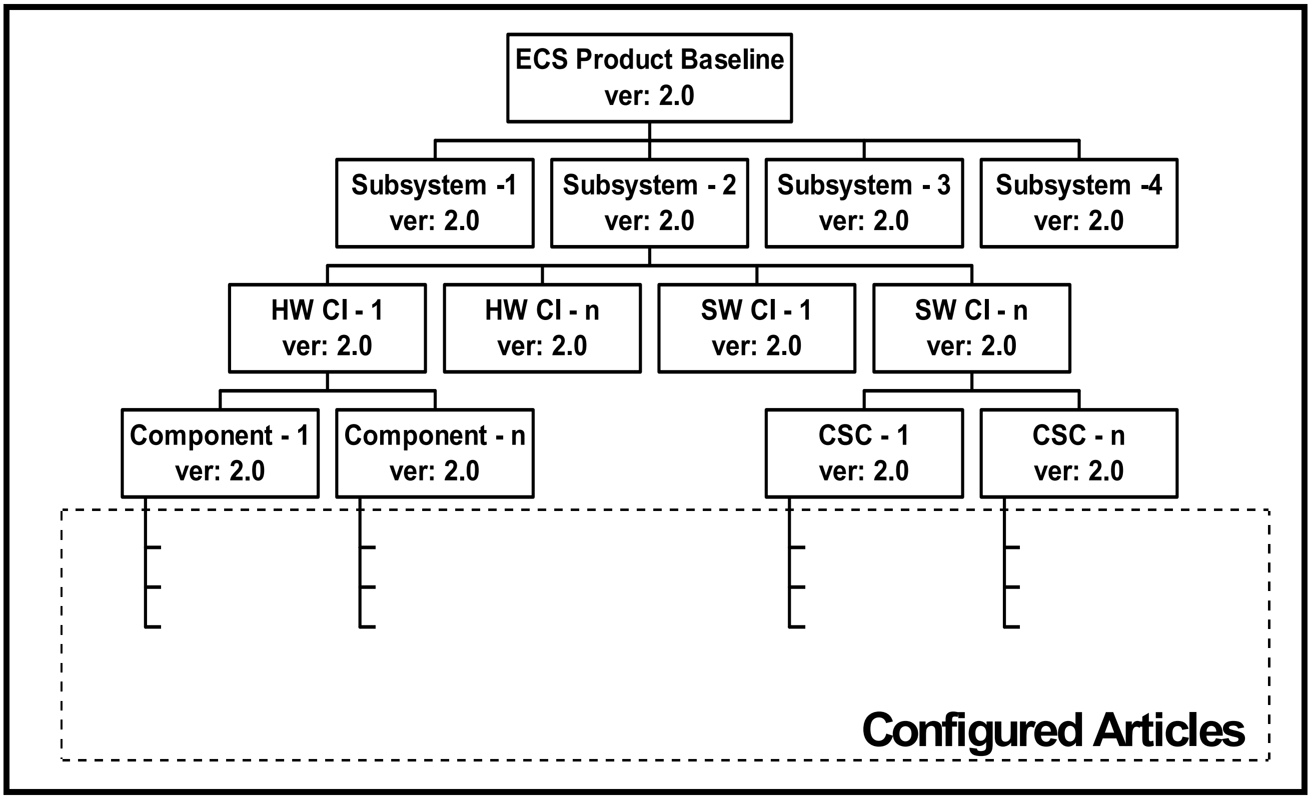


Figure 9.9.2-1. ECS Baseline Concept from a Design (CIL/CAL) View

At an operations site, the concept reflected in the upper layers of the Design View baseline structure may not be particularly useful. Although the same configured articles are involved, it may be desirable, for instance, to track items from the viewpoint of network administration. The resulting baseline product structure may reflect that shown in Figure 9.9.2-2.

Even if an operations site is to view ECS product structure as composed of subsystems, it is likely that the concept of CIs will be of little use. Instead, the site is likely to be focused on what hosts make up the subsystems. Therefore, the subsystem view at an operations site may be similar to that illustrated in Figure 9.9.2-3.

The Baseline Manager database implemented at the ECS Development Facility reflects ECS-developed product structures, and site personnel may not normally need all the data necessary to define these product structures. Instead, BLM tasks are likely to be limited to areas such as noting system changes, perhaps in the context of site-unique requirements and data. However, an understanding of the different ways of conceptualizing the system will help in interpreting baseline data reflected in the BLM.

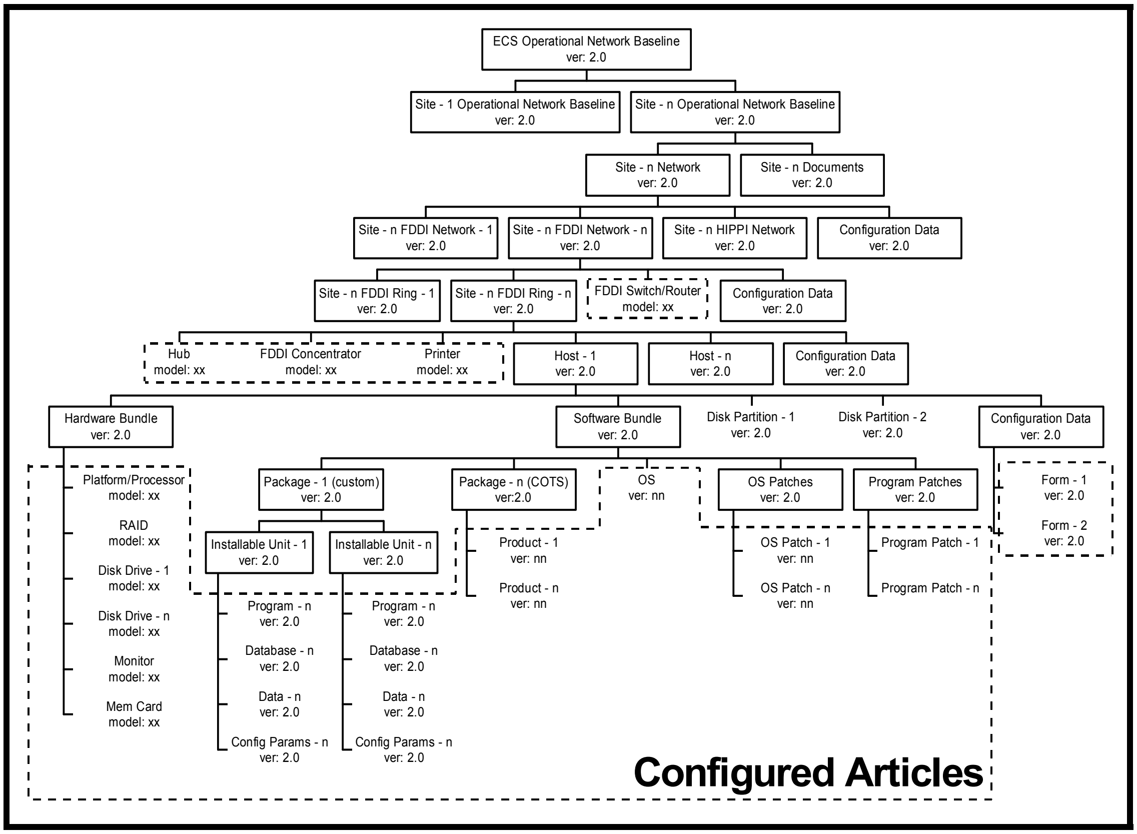

ECS Operational Network Baseline ver: 2.0
Site - 1 Operational Network Baseline ver: 2.0
Site - n Operational Network Baseline ver: 2.0
Site - n Network ver: 2.0
Site - n Documents ver: 2.0
Site - n FDDI Network - 1 ver: 2.0
Site - n FDDI Network - n ver: 2.0
Site - n HIPPI Network ver: 2.0
Configuration Data ver: 2.0
Site - n FDDI Ring - 1 ver: 2.0
Site - n FDDI Ring - n ver: 2.0
FDDI Switch/Router model: xx
Configuration Data ver: 2.0
Hub model: xx
FDDI Concentrator model: xx
Printer model: xx
Host - 1 ver: 2.0
Host - n ver: 2.0
Configuration Data ver: 2.0
Hardware Bundle ver: 2.0
Software Bundle ver: 2.0
Disk Partition - 1 ver: 2.0
Disk Partition - 2 ver: 2.0
Configuration Data ver: 2.0
Platform/Processor model: xx
RAID model: xx
Disk Drive - 1 model: xx
Disk Drive - n model: xx
Monitor model: xx
Mem Card model: xx
Package - 1 (custom) ver: 2.0
Package - n (COTS) ver:2.0
OS ver: nn
OS Patches ver: 2.0
Program Patches ver: 2.0
Form - 1 ver: 2.0
Form - 2 ver: 2.0
Installable Unit - 1 ver: 2.0
Installable Unit - n ver: 2.0
Product - 1 ver: nn
Product - n ver: nn
OS Patch - 1 ver: nn
OS Patch - n ver: nn
Program Patch - 1
Program Patch - n
Program - n ver: 2.0
Database - n ver: 2.0
Data - n ver: 2.0
Config Params - n ver: 2.0
Program - n ver: 2.0
Database - n ver: 2.0
Data - n ver: 2.0
Config Params - n ver: 2.0
Configured Articles

Figure 9.9.2-2. ECS Baseline Concept from an Operational (Network) View

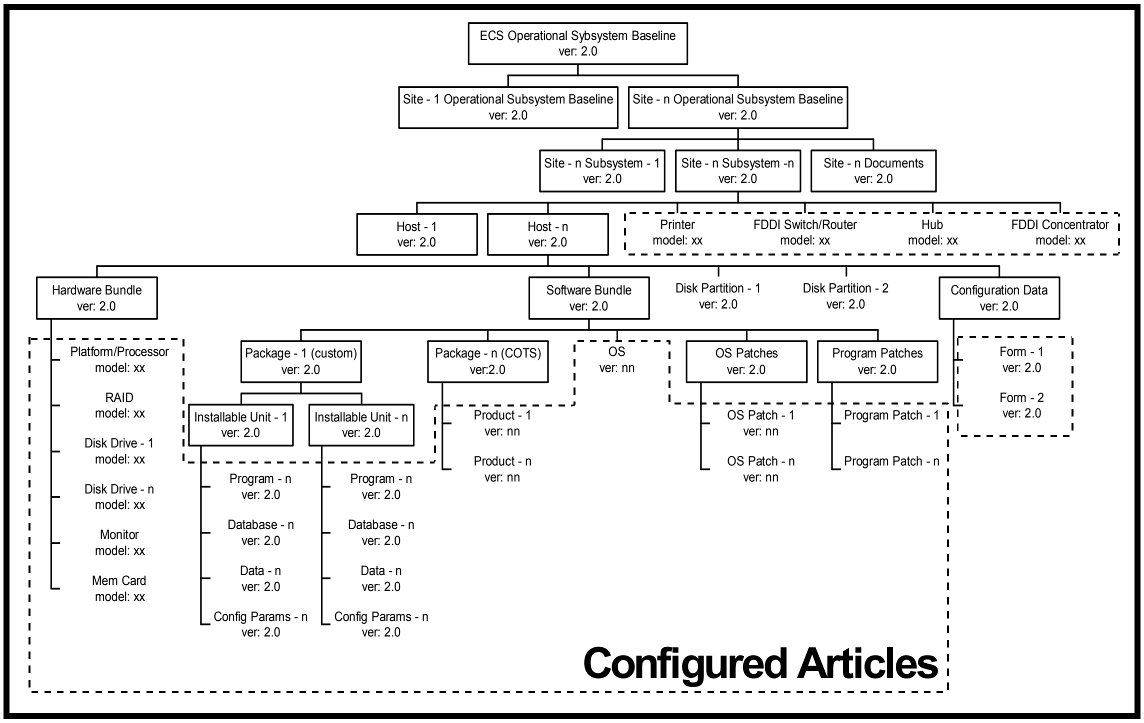

ECS Operational Sybsystem Baseline ver: 2.0
Site - 1 Operational Subsystem Baseline ver: 2.0
Site - n Operational Subsystem Baseline ver: 2.0
Site - n Subsystem - 1 ver: 2.0
Site - n Subsystem -n ver: 2.0
Site - n Documents ver: 2.0
Host - 1 ver: 2.0
Host - n ver: 2.0
Printer model: xx
FDDI Switch/Router model: xx
Hub model: xx
FDDI Concentrator model: xx
Hardware Bundle ver: 2.0
Software Bundle ver: 2.0
Disk Partition - 1 ver: 2.0
Disk Partition - 2 ver: 2.0
Configuration Data ver: 2.0
Platform/Processor model: xx
RAID model: xx
Disk Drive - 1 model: xx
Disk Drive - n model: xx
Monitor model: xx
Mem Card model: xx
Package - 1 (custom) ver: 2.0
Package - n (COTS) ver:2.0
OS ver: nn
OS Patches ver: 2.0
Program Patches ver: 2.0
Form - 1 ver: 2.0
Form - 2 ver: 2.0
Installable Unit - 1 ver: 2.0
Installable Unit - n ver: 2.0
Product - 1 ver: nn
Product - n ver: nn
OS Patch - 1 ver: nn
OS Patch - n ver: nn
Program Patch - 1
Program Patch - n
Program - n ver: 2.0
Database - n ver: 2.0
Data - n ver: 2.0
Config Params - n ver: 2.0
Program - n ver: 2.0
Database - n ver: 2.0
Data - n ver: 2.0
Config Params - n ver: 2.0
Configured Articles

Figure 9.9.2-3. ECS Baseline Concept from an Operational (Subsystem) View

9.9.3 Baseline Manager (BLM) Outputs Useful at the Sites

The BLM manages the COTS software, operating system patches, and COTS software patch baselines. The BLM records, including information on all scripts, data, and GUIs, are maintained and managed at the ECS Development Facility using ClearCase. The BLM tool produces some of the 910/920 Technical Document reports, with automated posting to ECS Baseline Information System (EBIS) and replication to server **cmdm.east.hitc.com**. The reports include the following documents that affect all sites:

- COTS Software Version 910-TDA-003.
- Site-Host Map 910-TDA-005.
- Critical COTS Software List 910-TDA-023.
- COTS S/W Where-Used Report 910-TDA-030.

The reports also include the following documents that are site specific:

- Hardware-Software Maps 920-TD(*x*)-002.

  (*Note*: The *x* represents a letter designating specific DAACs. *e.g.*, g = GSFC, l = LaRC, e = LP DAAC, and n = NSIDC.)

- Hardware-Patch Maps 920-TD(*x*)-014.

All BLM records are related to approved Configuration Change Requests (CCR) and Release Notes documents (i.e., series 914-TDA-xxx).

The Configuration Management (CM) organization is the principal user of the BLM tool, using it to implement changes to the baseline. The system is used daily to describe CCB-approved system components and to track sites and machines where version-controlled items are baselined. In addition BLM supports other functions such as configuration audits, system engineering and deployment activities. The BLM records describe the hosts for each site. The sites are the operational Distributed Active Archive Centers (DAACs), Verification and Acceptance Test Center (VATC) and Performance Verification Center (PVC). The system also tracks the COTS software and patches that are mapped to their respective hosts. The EBIS accommodates the identification of other baselined items such as documents, and disk partitions.

The BLM capabilities are used to:

- maintain records that identify what items comprise individual, baselined, system configurations
- identify the versions and variants of hardware and software items that are currently baselined together with the assemblies (e.g., hosts, subsystems, and networks) that use them
- record item interdependencies and the sites to which baselined items are deployed

- keep chronological histories of baseline changes and traceability of items to predecessor versions and system releases

9.9.4 Procedure for Retrieving Baseline Reports

When the ECS software baseline is changed (e.g., addition of a script, update or replacement of a Graphical User Interface (GUI) package), the change must be reflected in the collection, or "catalog," of control items that make up the affected Computer Software Component (CSC) assembly in the ECS product structure. In the BLM software at the EDF, to document the change it is necessary to add the new element to the catalog of version-controlled items, define an engineering change for the CSC assembly, and include the element in the list of items that will now make up that assembly. Once the change is documented, baseline reports reflect the new information. These reports may be accessed through the ECS CMDM server for the ECS Baseline Information System (EBIS). The EBIS page is obtained by a click on the **ECS Baseline** button at the end of the Tools line near the top of the ECS CMDM Server page. On the EBIS page, the ECS Baseline Information Technical Documentation is accessible through use of the **Technical Documents** button at the top of the row of buttons on the left side.

The resulting **ECS Baseline Technical Documentation** page lists the document series, title, and document number. The document numbers are links that provide access to the listed documents. The titles of some documents indicate BLM origin by inclusion of the parenthetical notation **(ClearCase)**.

The following procedure is applicable for retrieving baseline reports.

1. At the UNIX command shell prompt, type **setenv DISPLAY *clientname*:0.0** and then press the **Return/Enter** key.
   - For ***clientname***, use either the local terminal/workstation IP address or its machine name.
2. Start the log-in to a Netscape host by typing **/tools/bin/ssh *hostname*** *(e.g.*, g0ins02, e0ins02, l0ins02, n0ins02) at the UNIX command shell prompt, and press the **Return/Enter** key.
   - If you receive the message, **Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?** type **yes** ("**y**" alone does not work).
   - If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears; continue with Step 3.
   - If you have not previously set up a secure shell passphrase; go to Step 4.
3. If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, type your ***Passphrase*** and then press the **Return/Enter** key. Go to Step 5.

4. At the ***<user@remotehost>*'s password:** prompt, type your ***Password*** and then press the **Return/Enter** key.
   - You are logged in and a UNIX command shell prompt is displayed.
5. Type **netscape &** and then press the **Return/Enter** key.
   - The Netscape web browser is displayed.
6. Click in the **Netsite:** field.
   - The field is highlighted.
7. Type the Universal Resource Locator (URL) for the ECS Baseline Information Server (**http://cmdm.east.hitc.com**) and then press the **Return/Enter** key.
   - The **ECS CMDM Server** home page is displayed, offering access to ECS Baseline information as well as a number of tools, ECS web sites, and NASA EOS web sites.
8. On the Tools line, click on the **ECS Baseline** button.
   - The **ECS Baseline Information System** page is displayed
9. Click on the **Technical Documents** button.
   - The **ECS Baseline Technical Documentation** page is displayed.
10. Locate the desired report, scrolling down as necessary.
    - Reports derived from the BLM may be indicated with a parenthetical notation **(ClearCase)** in the title entry.
11. Click on the link for the document to be accessed.
    - A directory is displayed with one or more document numbers and versions indicated as links.
12. Click on the link for the document and version desired.
    - The document is displayed, and can be printed and searched.

This page intentionally left blank.

10. Metadata Administration

Every science data product generated and archived by the ECS must be described to the system by metadata which are put into an inventory and then used to retrieve and distribute the data to users of the system. The ECS Earth Science Data Model, described in documents 420-TP-020-001 and 420-TP-016, organizes the metadata into groups of related attributes and services to be performed on the data products. These "core" attributes are necessary to identify, interpret and perform services on granules and collections. The Data Model also provides for "product-specific" attributes, i.e. attributes which are unique to a specific data product.

The smallest aggregation of data that is independently described and inventoried in ECS is referred to as a data granule. Granules are organized into logical groupings called collections in which the granule metadata varies principally by time or location, called single-type collections.

Every collection is described by an Earth Science Data Type (ESDT) and is made known to the system by an ESDT descriptor file and associated software code which is built into the Data Server's dynamic link library (DLL) to perform the services.

Metadata administration includes creating and updating ESDTs within ECS. Collections may be modified and updated over time. In addition, quality assurance is performed after ESDTs have been installed and granules have been generated and stored in the archives. Collection-level metadata can be updated by updating the ESDT. Granule-level metadata can be updated manually (i.e. not as a result of an operation such as Subsetting which modifies the science data content of a granule) by setting the Quality Assurance flags and explanations. Procedures for updating these flags are provided in Section 15.

10.1 ESDT Descriptor Files

The primary task in establishing a collection is providing the core and product-specific metadata attribute values. This is done by creating an Earth Science Data Type (ESDT) descriptor file. The descriptor file is also used to specify the data services that are available for granules that belong to the collection. These services are implemented as methods of a Dynamic Link Library (DLL) containing C++ code to accomplish each service. The descriptor file and the DLL are the means by which a collection is made known to the Science Data Server (SDSRV).

The ESDT descriptor is composed of sections, all in ODL, containing the following information:

- Collection level metadata attributes with values contained in the descriptor.
- Granule level metadata attributes whose values are supplied primarily by the Product Generation Executives (PGEs) during runtime.
- Valid values and permitted ranges for all product-specific attributes.
- List of services for all the granules in the collection and events which trigger responses throughout the system.

The ESDT descriptor file is created by Metadata Works (MDWorks), discussed in Section 10.3.3.

The services that apply to a collection are specified in the ESDT descriptor file. Metadata Works automatically inserts standard ECS-supplied services such as insert and search into the descriptor file. Product-specific services, such as Subsetting or a product-specific acquire, require executable code to enact those services. This code is contained in the DLL. The DLL is written and tested by either the ECS developers or sustaining engineering personnel at the DAAC.

After the ESDT (both descriptor file and DLL) has been generated it must be installed on the Science Data Server before the first data granule can be inserted. During this installation process, information from the ESDT Descriptor File is propagated to the Data Management and Interoperability subsystems, and to the Subscription Server, which must all be operating during the ESDT installation process. The detailed procedures for ESDT installation into ECS are described in Section 11.3.

10.1.1 Steps in Generating a Descriptor File

ESDTs for Distributable Product

These are the typical steps used in generating a descriptor file:

.Identify desired collection-level metadata attributes

- For permanent and interim files use only the minimum attributes.
- For distributable products identify all applicable attributes. This will involve reading appropriate documentation and interacting with the data provider.

.Identify granule-level attributes

- If a sample metadata configuration file is available from the data provider, use this.

.Check valids for core attributes (write CCR if new valids are required).

.Check PSAs (register PSAs if new).

.Gather metadata into a spreadsheet, or use Metadata Works to enter metadata directly.

.Use custom built scripts to generate the descriptor file from the spreadsheet, if used.

.Verify the descriptor file as outlined below.

.Check descriptor files into ClearCase.

.Notify the DLL Team Lead of the newly prepared descriptor files.

10.1.2 Verifying Descriptor Files

- Run the PERL script "update.pl", following the instuctions in the script prologue. (This script makes sure that the inventory metadata attributes are all listed as event qualifiers in the EVENT group.)
- Run the PERL script "esdtQC.pl", following the instructions in the script prologue. Make any necessary corrections in response to errors issued, and rerun. Repeat until there are no errors. (This script checks for more than 30 common descriptor file errors.)
- Run the PERL script "required.pl", following the instructions in the script prologue. Add any missing attributes as indicated.
- Run the "testodl.csh" utility to ensure that there are no errors in the ODL structure for the descriptor file. Make any necessary corrections in response to errors issued, and rerun. Repeat until there are no errors.

10.2 Preparation of Earth Science Data Types

An ESDT goes through pre-operational life cycle steps starting with an analysis of the collection's need and continuing through development and operational installation. This process involves actions by the Data Provider or User in addition to ECS. These procedures are detailed in Project Instruction SO-1-002, "Earth Science Data Type Generation Procedures". The overall workflow is shown in Figure 10.2-1.

DEFINITIONS

Archive - A File Type indicating granules will be inserted to Data Server for long term storage and acquisition for distribution.

Full - A level of metadata coverage intended for data products which are produced within ECS.

Collection - A related group of data granules.

Granule - The smallest data element which is identified in the inventory tables.

Interim - A File Type indicating granules are temporarily stored in support of product generation.

Intermediate - A level of metadata coverage intended for contemporaneous data products which are not produced within ECS.

Limited - A level of metadata coverage intended for heritage data products brought into ECS for distribution

Minimal -A level of metadata coverage sufficient to uniquely identify a collection or granule.

Permanent - A File Type indicating static or semi-static granules which are used only as inputs in product generation.

Product Specific Attributes - Attributes that are defined by the data provider in support of searching for specific granules

Valid - An allowable metadata value.

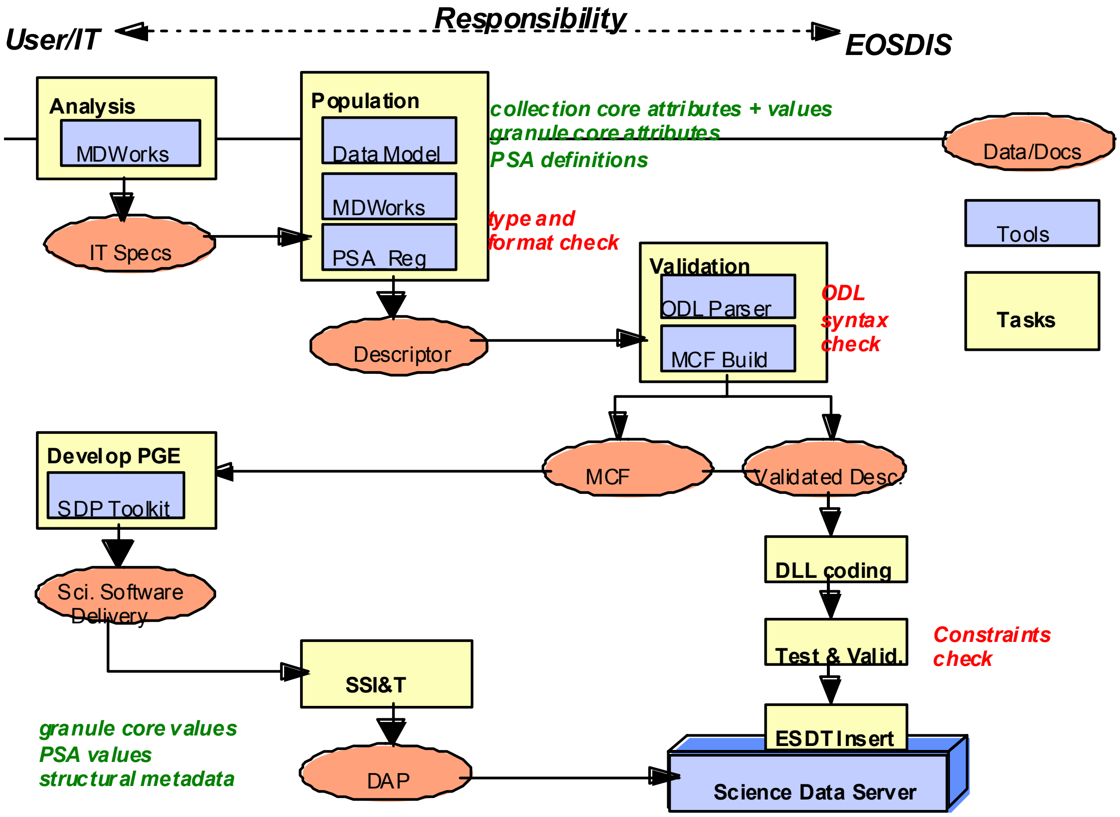


Figure 10.2-1. The ESDT Generation Process

PROCESS:

1. Need Analysis

- The baseline list of science ESDTs and their services is controlled by the ECS CCB. This baseline was established through an analysis of the ECS Functional and Performance Requirements Specification, the Technical Baseline established from inputs from the Ad Hoc Working Group on Production, and meetings with the individual data providers to define the basic requirements of each ESDT.

  These basic requirements are:

  - Data Provider File Designation,
  - File Type (Permanent, Interim, Archive)

    - Level of Metadata Coverage (Minimal, Limited, Intermediate, Full)

- For new ESDTs not currently in the development baseline, the result of the Need Analysis forms the basis for approving the inclusion of the ESDT into ECS. This is accomplished through the CCR process, governed by PIs "Configuration Change Request Preparation" [CM-1-003] and "Change Control Board Process" [CM-1-004].

2. ESDT Specification

- This step results in a set of specifications extending the results of the needed analysis and providing the information needed to implement an ESDT. This step is executed only if the ESDT has been included in the baseline. The roles and responsibilities for developing the specification are as above.

  The specifications must include:

    - ShortName and VersionID of the ESDT
    - A list of the metadata attributes needed, valids, and any constraints on attributes. This list is to be drawn from the B.x Descriptor File Template. This template, under CCB control, is based on the attributes defined in "Release-B SDPS Database Design and Database Schema Specifications" [311-CD-008-001] (i.e., DID 311), as modified for B.0 (for example) by "B.0 Earth Science Data Model " [420-TP-020-001].
    - A list and specification of the services needed (e.g., specification of the INSERT, SEARCH, ACQUIRE and SUBSCRIPTION semantics).

3. ESDT Generation

- Once the ESDT Specification has been developed and the applicable attributes identified, the necessary metadata has to be gathered, the metadata values checked against the valid values and the product-specific attributes (PSA) need to be checked against the list of PSAs that are already defined (see Fig. 10.2-2).

- Once the collection-level metadata and granule-level attributes have been checked, then the descriptor file is generated, the Dynamic Link Library produced, and testing and validation of the ESDT performed. This process is further elaborated in the sections below.

- For a one-of-a-kind, distributable product with Full metadata coverage, this process can take up to six weeks to accomplish. For a related group of products with identical services, much of the Descriptor File and DLL of the first ESDT can be reused, and the cycle time for preparing subsequent ESDTs in the related group is much less.

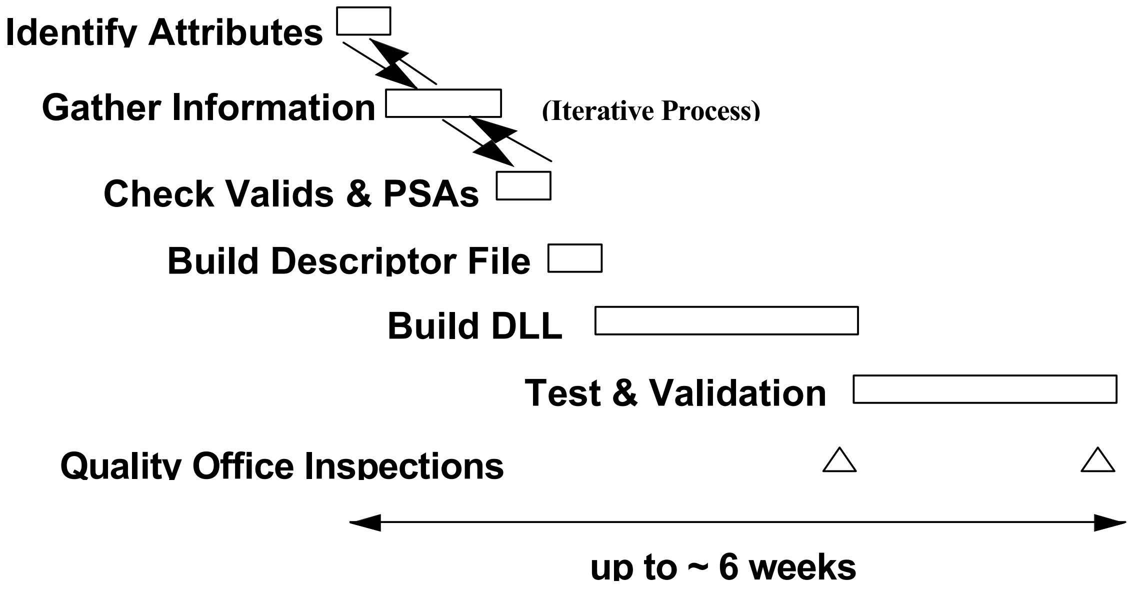


Figure 10.2-2. Steps in ESDT Development

10.3 Tools used in Generating a Descriptor File

Figure 10.3-1 shows the some of the tools that have been developed, and indicates supporting information flows between these tools. Brief descriptions of these tools are provided below.

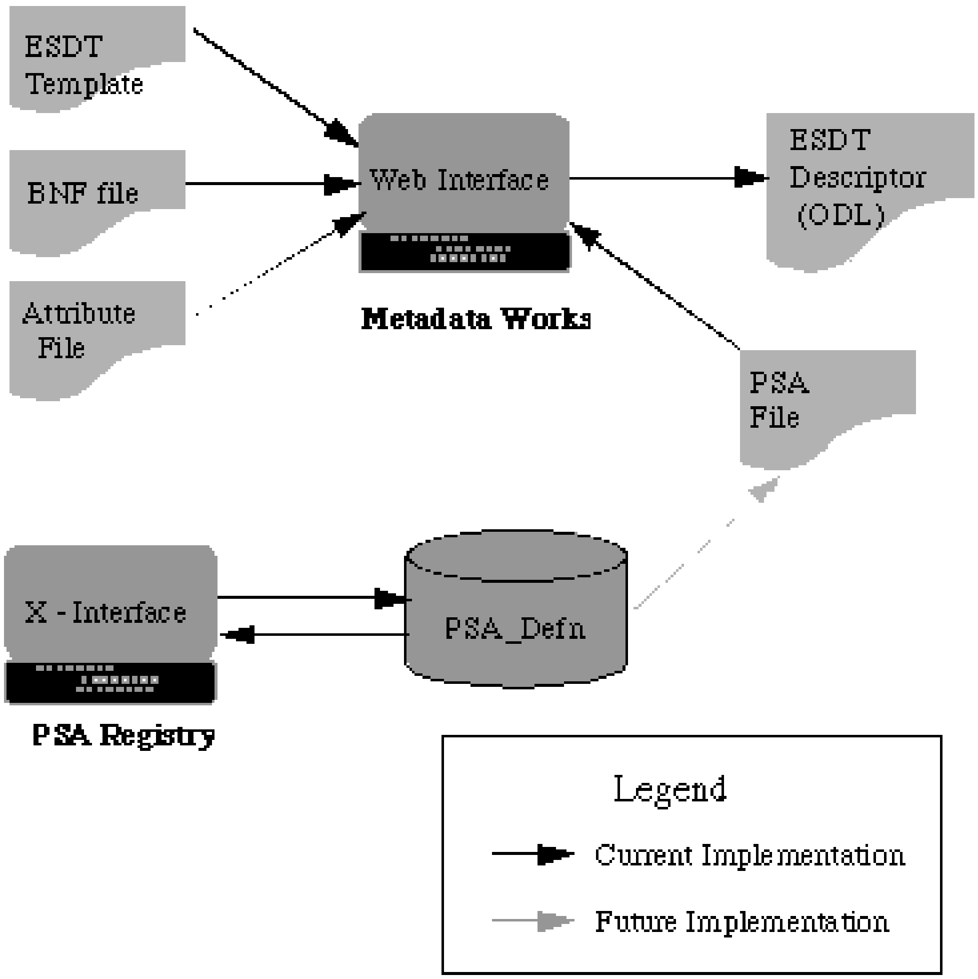


Figure 10.3-1. Tools and Supporting Information Flows

10.3.1 Data Dictionary and Valids Checking

Each proposed new valid must be reviewed by the Data Modeling group in order to validate that they fit within the model. When approved, within a couple days at most, the new valid will be available for use.

10.3.2 PSA Registry

Each Product Specific Attribute must be registered to ensure that the name is uniquely defined across all products and that no two names are applied to the same definition (aliases may be applied at the user's Client and in Data Management, but should not be applied within the inventory). An X-Windows based GUI has been developed to assist in the registration process.

10.3.3 Metadata Works

To support the entry of metadata for ESDTs with Full or Intermediate coverage, an HTML-based GUI has been developed. Metadata Works allows the person entering the metadata to start with a completely empty form, or to define a new descriptor file employing defaults of already defined groups or entire previously defined descriptor files. When the metadata entry is complete for a descriptor file, Metadata Works is then used to generate a complete Descriptor file employing the proper ODL syntax.

10.4 Metadata Population

10.4.1 Collection-Level Metadata

A majority of the attributes in the ECS Data Model apply to all the granules in the collection. These are known as collection-level attributes. There can be both core and product-specific collection-level attributes, defined once prior to establishing the collection.

Collection-level metadata is input via an HTML-forms GUI tool called Metadata Works, described further in Section 10.3.3. Based on the ECS Data Model it is designed for use by the data provider; e.g., an instrument team scientist or other person having extensive knowledge of the data. It can import as defaults the attribute values from a collection that has already been populated. A sequence of screens is presented to the user enabling specification of all required and optional attributes, with a list of permitted values presented where appropriate. Help screens give attribute definitions, data types and other relevant information to assist in the specification.

10.4.2 Granule-Level Metadata

The attributes in the ECS Data Model which can vary on a granule-by-granule basis are known as granule-level attributes. There can be both core and product-specific granule-level attributes.

Granule-level metadata are specified and populated using the Metadata Configuration File (MCF). The MCF is derived from information contained in the ESDT descriptor file and delivered by the Science Data Server for use by the Science Data Processing or Ingest Subsystems. The MCF specifies how the searchable metadata attributes will be populated in the SDSRV database. For data products generated by ECS, the science software or Product Generation Executive (PGE) interacts with the MCF using metadata tools contained in the Science Data Processing Toolkit. Through this process values are set for metadata attributes specified in the "source" MCF, such as the temporal or spatial coverage of each granule. These values are then inserted into a "target" MCF at PGE run time. The MCF is used in a similar manner for data entering ECS through the Ingest subsystem.

Procedures for entering data into ECS through the Ingest subsystem are described in Chapter 16. "Ingest". Procedures for running a PGE are described in Chapters 11.10 "Running a PGE in a Simulated SCF Environment at the DAAC" and 11.13 "PGE Planning Processing and Product Retrieval".

The Inventory Metadata section of Metadata Works is used to capture granule-level metadata specifications. An unofficial MCF may be generated as output from Metadata Works, for testing purposes. Final testing should always be done with the MCF provided by ECS which is guaranteed to be identical to the one delivered by SDSRV at run time.

The actual population of the granule-level attribute values into ECS inventory data bases takes place during the insert of a data granule into the SDSRV. Each data granule consists of one or more physical files. Accompanying each granule is a metadata record; i.e., an ASCII file containing the granule level attributes and their values in ODL. Only one metadata record is allowed per granule, i.e. no. sub-granule records are allowed, and no metadata records are shared between granules.

Procedures used to initiate the running of PGEs are described in Chapter 11.13 "PGE Planning and Processing".

10.4.3 Product-Specific Metadata

Product-specific metadata can be at both the granule-level and the collection-level. Product-specific metadata may (at the data providers election) be contained in the inventory tables in the database, in which case it will be searchable by ECS. There is also a provision to store product-specific metadata within granules that is available only when the granule has been ordered and delivered. This is termed archive metadata and is specified in a separate ODL group in the MCF.

In the granule metadata, the core attribute that is available to store product-specific metadata is called ParameterValue. The metadata describing this attribute is specified by the data provider through the AdditionalAttributes class at the collection-level. The units of measure, range, accuracy and resolution for this is specified in the PhysicalParameterDetails class, also at the collection-level.

Product-specific metadata at the collection level is specified with Metadata Works at the time the other collection level metadata attributes values are defined. At the granule-level, product-specific metadata is defined in the MCF. In both cases, a list of valid values and permitted ranges are specified in the ESDT data dictionary.

10.5 Testing and Validation

Testing and validation involves installation of the ESDT on a Data Server, and subsequent tests of the data services for the ESDT. These tests include insertion of actual or simulated data, search acquire, and other services that may apply and be available and supported under the extant version of ECS. (Section 11.3)

After testing, the ESDT Descriptor File and DLL are promoted to the development CM environment if pre-ECS release, or to the operational environment if after the ECS release is made operational.

This page intentionally left blank.

11. Production Rules for PGE's to Function in ECS

11.1 Production Rules

11.1.1 Purpose and Scope

This section describes the production rules supported by the Release B CDR design. It is intended to explain the implementation of these production rules in ECS and to document the information required for each rule. This section does not cover the exact design and implementation of these production rules in the PDPS system. That information may be found in the PDPS design documents, primarily in the Planning Subsystem document 305-CD-026-002, Release B SDPS Planning Subsystem Design Specification for the ECS Project.

While this section does address the syntax and operational procedures for specifying production rules, it does provide detailed information about what data is required for each rule and how that data will be used.

11.1.2 Overview of Production Rules

11.1.2.1 Data Processing in ECS

Before discussing production rules, it is important to have a basic understanding of how the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) conducts its data processing. ECS provides facilities for running product generation executives (PGEs) to produce science data products. The support for this function primarily resides in three subsystems: the Data Processing Subsystem (DPS) which actually executes the PGEs, the Planning Subsystem (PLS) which plans the execution of PGEs to efficiently utilize the local computing resources and the Data Server Subsystem (DSS) which provides PGEs with input data and archives output data. The design and implementation of the DPS and the PLS are carried out by the ECS organization known as the Planning and Data Processing System (PDPS).

Data Processing Subsystem (DPS) - The DPS provides the environment under which PGEs are run. It works with the SDP toolkit to interface with the science software. The DPS coordinates the acquisition of input data (staging) and the archiving of output data (de-staging) with the DSS.

The DPS also ensures that there are adequate resources (disk space, RAM, etc.) available before a PGE begins execution. The DPS uses the AutoSys COTS product to implement the PLS production plan.

Planning Subsystem (PLS) - The PLS generally manages the ECS data processing activities and provides the main interface between the DPS and the rest of ECS. The PLS plans the execution of PGEs to provide efficient use of computing resources. The PLS receives. Production Requests (PRs) either from a software tool (Production Request Editor) or, for on-demand production requests (OPRs), from the DSS or DPS (see Error Handling). The PLS breaks up PRs into data processing requests (DPRs), which are individual runs of a PGE. It maintains information about input data for specific DPRs and passes that information to the DPS. It queries the DSS for data which meets required criteria and provides the universal references (URs) of that data to the DPS for acquisition.

Data Server Subsystem (DSS) - The DSS provides input data to the DPS and archives output data from the DPS. It responds to queries from the PLS and provides the PLS notification of the arrival of input data for all subscriptions the PLS has placed with it. The DSS also submits On-Demand

Production Requests (OPRs) to the PLS.

These three subsystems form the core of the ECS data processing services. Since it has responsibility for managing these activities, most of the information about production rules will be maintained by the PLS.

11.1.2.2 General Definition of Production Rules

Simply stated, production rules are the instructions about how a particular PGE is to be run.These instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions. This section focuses on the other types of production rules such as alternate inputs to be used or conditional PGE activations. These and other production rules are defined in the following sections.

Production rules in the ECS system will be tied to PGEs. This means that each PGE will use one or more production rules. The production rules will be initially entered by the Instrument Team with further modifications if necessary when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values will be entered at that time. Many of these defaults can be overridden in the production environment when a Production Request is entered.

11.2 Production Rule Definitions

Introduction

This section presents a basic definition and examples of each of the production rules currently supported by the Release B CDR design. The production rules are divided into three categories:

Input Data Specification

Conditional Activation

Error Handling

Input Data Specification rules specify how the inputs for a particular run of a PGE are identified. Conditional Activation rules stipulate the conditions of when a particular PGE is to be run. Error Handling rules specify automatic actions to be taken when a PGE fails. Each sub-section is titled with the name of the production rule being discussed.

It is hoped that providing a common nomenclature will facilitate any future discussions between ITs and ECS about production rules. After each rule there is a list of the information required to implement that rule in the production environment (i.e. post-launch, at the DAACs). This information will initially be provided at SSI&T time. In the operational environment many of the values can be set when a Production Request is entered. Most production rule sections also include a simple diagram to help illustrate the concept being presented.

11.2.1 Input Data Specification

These production rules generally stipulate how inputs for a particular run of a PGE (a DPR) are identified/selected/created. Most PGEs have at least one input and one output. The most basic information about these inputs and outputs is the Earth Science Data Type (ESDT). It is assumed in the examples in this chapter that the ESDT is specified for all data sets which are used or produced by a PGE.

11.2.1.1 Basic Temporal

The most basic type of production rule specifies, for each input ESDT, the temporal range of that ESDT required by the PGE.

For example, the run of a PGE which produces a particular 2.5 minute MODIS Level 1B data granule requires as input the specific 2.5 minute Level 1A granule which is ontemporaneous with the Level 1B granule to be produced.

Another example would be the production of a CERES Level 1A granule from the Level 0 data. CERES L1A granules cover 24 hours and require the PLS to identify and the DPS to stage all of the Level 0 data files that cover a particular 24 hour period. This sort of simple temporal specification is basic to the system and will be used by every PGE.

Inputs required to implement the Basic Temporal rule

For each PGE which uses this rule, the following data must be provided for each ESDT (input oroutput): Time period - Length of time of ESDT (e.g. 12 hours, 1 week or one year).

Boundary - Date/time to start counting periods

Both of the above parameters are specified at SSI&T time.

11.2.1.2 Example use of the Basic Temporal Rule

Suppose there is a PGE which creates a daily (24 hour) data set and those data sets should correspond with calendar days. In that case Time Period = 24 hours and Boundary = 1/1/98 00:00 (this could be any date). When a Production Request is entered, the Planning Subsystem will scan backwards from the start time of the PR until it finds the start of a period. It then produces DPRs for each period of time covered by the PR. Table 11.2.1.2-1 illustrates how this would work. Given the period and boundary defined above, two PRs are entered, the first for the time between 6/3/99 12:00 - 6/7/99 6:00 while the second specifies 6/7/99 18:00 - 6/16/99 12:00. When the first PR (#1) is entered, the Planning Subsystem determines that the start time of the PR is not the start time of a period. It then scans backwards, finding the start time of the period to be 6/3/99 00:00 and creates a DPR (1.1) to process that period. It then continues producing DPRs (1.2, 1.3, 1.4 and 1.5) until it reaches the end of the period which contains the end time of the PR. When the second PR (#2) is entered, the same process takes place. Note that since the end time of PR #1 and the start time of PR #2 fall in the same period, duplicate DPRs (1.5 and 2.1) are created. This sort of duplication is checked for by the Planning Subsystem and the second DPR is not run.

Table 11.2.1.2-1. Basic Temporal

Boundary (1/1/98 00:00)
= Period (24 Hrs)
PR #1 PR #2
DPR 1.1
DPR 1.2
DPR 1.3
DPR 1.4
DPR 1.5 = DPR 2.1
DPR 2.2
DPR 2.3
DPR 2.10
DPR 2.9
6/1/99 6/23/99

11.2.1.3 Alternate Inputs

This is a fairly simple rule, although there are several options which give it great flexibility and power. There are cases where, for one of its input data sets, a PGE can use any one of several inputs. For example, a particular PGE might require model wind data as an input and would be capable of accepting wind data from a DAO model, an NCEP model or, as a last resort, could use climatology. Variants of this rule allow the alternate input to be optional or allow alternates to be grouped so that more than one of the alternates may be needed (i.e. use any M of N ESDTs in the group).

11.2.1.4 Inputs required to implement the Alternate Inputs rule

ESDT of Primary Alternate - This is the data set ID for the most desirable alternative.

Timer for Primary Alternate - Amount of time to wait before attempting to use second alternate. Number Needed - Number of the alternate data sets (from this group of alternatives) required by the PGE. Usually 1, but can be more. If set to 0, this set of alternates is optional (i.e. the PGE can run without any of the alternates) ESDT of Second Alternate -

Data set ID for Second Alternate

Timer for second Alternate - Amount of time to wait before attempting to use third alternate.. (include any number of alternates, no practical limit).

ESDT of Last Alternate - Data set ID for Last Alternate.

Timer for Last Alternate - Note that this timer is used only if Number Needed = 0, which indicates that this set of alternates is optional. In that case, if this timer expires, the DPR will be executed even if none of the alternates is available. If however, Number Needed > 0, then this timer is not used; the DPR will be held indefinitely, waiting for one of the

alternates to become available.

11.2.1.5 Example use of the Alternate Inputs Rule:

Let's assume that a PGE has two required input data sets and a third data set which can be any one of three alternates. Of the three alternates, the first two choices are almost equivalent, while the last choice is less desirable. In this case we designate the Primary Alternate (first choice) and set a timer for 1 hour. The second choice is designated with a timer set for 4 hours. This situation is shown in Table 11.2.4-1. What will happen is that, after the two required inputs are available, an attempt is made to acquire the first choice. If that choice is unavailable, the Planning Subsystem holds the DPR and starts the first timer (1 hour). If the first choice becomes available in that hour, the DPR is started immediately. If the first timer expires and the first alternate is still unavailable, an attempt is made to use the second choice. If the second choice is unavailable, the second timer (4 hours) is started. If the first or second alternates become available at any time during that 4 hours, the DPR is started immediately. However, if the second timer expires (a total of 5 hours after the two required inputs are available), an attempt is made to use the third choice. Unless this set of alternates is optional, most PGEs are expected to have a last alternate which is always available (e.g. climatology). If not, the DPR is held indefinitely waiting for one of the alternates to become available.

Table 11.2.1.5-1. Alternate Inputs

Required Dataset 1
Primary Alternate
Second Alternate
DPR
< or > First Choice –Timer set to 1 hour.
Wait 1 hour after required data sets are availabe before trying to use second alternate.
Second Choice –Timer set to 4 hours.
Wait 4 hours after this choice was first tried before attempting to use third alternate.
Output Dataset
Third Alternate < or > Third (in this case, last) choice. Attempt to use if primary and secondary alternates are unavailable 5 hours after required datasets are available.
Required Dataset 2

11.2.1.6 Tiling

In this case a PGE is set up to run for a series of pre-defined, rectangular areas (tiles). The tile definitions need to be entered into the PDPS database so that the proper input data granules can be identified. The implementation of this rule calls for the PGE developers to create tile definition files which would have descriptions of each tile. The Planning subsystem would then use those definitions to query the Data Server for input data granules for each tile. Tiles are grouped together into clusters which are likely to share common input granules. This is done for efficiency of operations.

11.2.1.7 Inputs required to implement the Tiling rule:

The primary input for the tiling rule is a tile definition file. This file will contain one entry for every tile which will include: Tile ID - Unique identifier used to refer to the tile Lat./Lon. of tile corners. Four coupled pairs of coordinates which bound the tile. Cluster ID - ID of cluster in which tile falls.

Table 11.2.1.7-1. Tiling (1 of 2)

Tile 1 Input Data
Tile 2 Input Data
Tile 3 Input Data
Tiles 4-6
Tiles 7 and 8
DPR-1
DPR-2
DPR-3
Output granule for tile 1
Output granule for tile 2

Table 11.2.1.7-1. Tiling (2 of 2)

Output granulefor tile 3
Tiles 4-8 are processed in the same manner as tiles 1-3.
8 Tiles clustered together since they are likely to have overlapping inputs
Original
Swath
Data

11.2.1.8 Example use of the Tiling Rule

A tile definition file is part of the package delivered with the PGE for SSI&T. In the production environment, a PR is entered to run the PGE on all the tiles. A DPR is generated for each tile. These DPRs are grouped by Cluster ID so that, since they will share some of their inputs, these DPRs will be run on the same machine which will minimze the the staging and copying of those inputs amongst the various production machines. The PLS will use the tile definitions to query the DSS to idenify all input data granules covering a tile. It is likely that many of these input granules will have data outside as well as inside each tile. The URs for these input granules and the specific tile definition are then passed to the DPS as part of each DPR. Each DPR will then run, the DPS will aquire the inputs from the DSS and the PGE will run and, using only the data in the tile, create an output data granule for its tile.

11.2.1.9 Data Server Proxy (Subsetting/Subsampling)

There are cases where a PGE would like to use Data Server services on its input data sets prior to running the PGE. In these cases the Planning and Data Processing subsystems simply act as proxies for the PGE by calling those Data Server services. The services provided here are totally dependent what services the DSS offers for each specific product. Instrument teams wishing to use specific services on specific products should coordinate with the Data Server Subsystem to ensure those services will be available.

11.2.1.10 Inputs required to implement the Data Server Proxy rule

Each PGE which uses this rule will be need to identify:

Input ESDT - ID of data set upon which Data Server services will be performed.

Service Requested - Data server service ID Interface Parameters - Parameters specific to the service being requested. For example, a spatial subset request would specify the geographic extent while a temporal subset request would specify the time period requested.

Table 11.2.1.9-1. Subsetting & Subsampling

Subset of Original Dataset
DPR Output Data Granule
DPR Output Data Granule Subsample of Original Dataset
Spatial Subsetting Subsampling

11.2.1.11 Example use of the Data Server Proxy Rule

Assume a PGE with an input data set which covers the entire globe and is gridded at a .1 o by .1 o resolution. During the post-launch, PGE shakedown phase, the science software developer might want to generate a 1 o by 1 o resolution product for a quick look at the output data. Rather than have the PGE do its own averaging/subsampling, Data Server subsampling services could be invoked so that the PGE only has to run on 1/100th the amount of the original data. In this case a second PGE profile would be created at SSI&T which would include the service request and track the different runtime characteristics (e.g. CPU time, disk storage for input/outputs, RAM usage, etc.)

11.2.1.12 Level 0 to Level 1A

This is a special case of the simple temporal range rule and is of interest only to certain Level 1A PGE developers. The issue is what rules PDPS uses to determine what Level 0 data files are required to produce a particular Level 1A granule. In many cases (such as CERES), this is handled by the Basic Temporal rule rather than the method described here. This rule is intended to handle orbit-based Level 1A granules. Level 0 'granules' received by EDOS will be not have any discernible relationship with the spacecraft orbit. The Level 0 'granules' have not even been fully defined for some instruments flying on other platforms. In these cases, the Planning Subsystem will access a crude orbit model (really just a lookup table) to determine the start/stop times of a given orbit. It will then use those start/stop times (after adding a small cushion to ensure complete coverage of an orbit) to query the data server for the Level 0 data covering the orbit. The response to that query is given to the Data Processing Subsystem which acquires the data. The PGE then uses SDP Toolkit calls to determine the exact extent of the orbit. A similar process will be used for instruments having orbit-based Level 1A granules but which will be flying on platforms other than AM-1. In the case of SeaWinds, flying on the ADEOS II platform, a temporal offset may have to be used if the orbit model is based on equator crossing rather than on pole crossing as the SeaWinds want for their Level 1A granules.

Table 11.2.1.12-1. Level 0 to Level 1A

Time
Level 0 Data
Orbit-based
Level 1A Granules
Using temporal definitions of orbits, the PLS identifies and the DPS stages the appropriate Level 0 Data for each orbit-based Level 1A granules

11.2.2 Conditional Activation:

Most PGEs have well defined times and conditions when they are to be executed. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g. run once for every granule or run monthly averages once a month). However, some PGEs might have additional/different constraints on when they get run. This section addresses those cases.

11.2.2.1 Intermittent Execution

A PGE can be set up to run on every Nth instance of input data. For example, a QA PGE that is run on a daily product may only need to be run every fifth day to provide a spot check. Note that this does not refer to the common case of only running a weekly averaging PGEs once each week, which would be handled by the Basic Temporal rule and the time ranges specified for the input and output ESDTs. Rather this is a special case where a PGE can be run every day (or hour, week, etc.), but, for whatever reason, it is only desired to be run every Nth day.

11.2.2.2 Inputs required to implement the Intermittent Execution rule

This rule is implemented by using two parameters:

Number to Skip - Number of DPRs to be skipped (not executed)

Number to Keep - After skipping the specified number of DPRs, how many are to be kept. This number will usually be one, but could be any number. The use of these parameters will allow a pattern of execution to be established. This pattern is not maintained between PRs.

Table 11.2.2.2-1. Intermittent Execution (1 of 2)

PR #2
PR #1
DPR
1.1
DPR
1.2
DPR
1.3
DPR
1.4
DPR
2.1
DPR
2.2
DPR
2.3

Table 11.2.2.2-1. Intermittent Execution (2 of 2)

DPR
2.4
DPR
2.5
DPR
1.2
DPR
1.3
DPR
2.3
DPR
2.2
DPR
2.5
PRs #1 and #2 are entered at different times.
PR #1generates 4 DPRs and PR #2 Generates 5.
Only 5 of the 9 are ‘kept’and actually run.
Number Skipped = 1
Number Kept =2
Created Run
Separate PRs,
pattern is restarted for PR#2

11.2.2.3 Example use of the Intermittent Execution Rule

Using the above example of a QA PGE to be run every fifth day, let's assume a Production Request is entered which covers the period from 6/1 - 6/30. As part of the PR values are entered for number skipped (4) and number kept (1). The PR would be expanded into 30 daily DPRs, of which, four out of every five would be discarded, leaving one DPR every fifth day.

11.2.2.4 Metadata-based Conditional Activation

It is possible to determine if a given DPR should be run, based on the metadata of one or more of its input data sets. For example, a PGE could be set up so that a QA flag must be set to an acceptable level/state within the metadata of an input data set or the PGE should not be run. This production rule will work for both core and product-specific metadata. Note that this is different than data-based conditional activation, which will not be supported in Release B. If that sort of conditional activation is desired, the IT will need to define a product-specific metadata field which will be filled by the PGE producing that data.

11.2.2.5 Inputs Required to Implement the Metadata-based Conditional Activation rule

This rule is implemented by using a series of statements for each ESDT to be checked. These statements take the basic form of:

Metadata field Operand Value. These statements would be 'AND'ed together so that if any of the checks fail, the DPR will not be run. These statements would be entered in at SSI&T time, however, the values could be changed when a Production Request is entered.

Table 11.2.2.5-1. Metadata-based Conditional Activation

Input 1
QAFlag=7
DayNight='DAY'
Clouds=20%
Input 2
QAFlag=4
Input 1
QAFlag=9
DayNight='DAY'
Clouds=70%
Input 2
QAFlag=8
Input 1
QAFlag=7
DayNight='DAY'
Clouds=30%
Input 2
QAFlag=7
DPR 1
DPR 2
DPR 3
Metadata Checks:
Input 1:
QAFlag > 5
DayNight = 'Day'
Cloud < 40%
Input 2:
QAFlag > 6
DPR 1 is NOT run since Input 2 failed the QAFlag check.
DPR 2 is NOT run since Input 1failed the cloud cover check.
DPR 3 IS run since Inputs 1 and 2 met all metadata conditions

11.2.2.6 Example Use of the Metadata-based Conditional Activation Rule

Assume there is a PGE which uses multiple ESDTs as inputs. Two of the inputs (Input 1 and Input 2) need to have their metadata checked prior to the PGE being executed. Input 1 needs to have a certain quality, less than a certain percentage of clouds and be daytime data while Input 2 just needs to be of a certain quality. Table 11.12.3.4-1, illustrates this situation and shows three DPRs created for different instances of Inputs 1 and 2 and the disposition of those DPRs based on the metadata of the inputs. As stated earlier, if it was decided that the cloud cover rule was too strict, the value used for comparison could be changed when Production Requests are entered. If, however, a new check were needed for some other metadata field, this change would have to be done through SSI&T.

11.2.2.7 Mode-based Conditional Activation

In this case, the mode a given instrument is in will determine which PGE is run. For example an instrument might go into a calibration mode which requires that a special calibration PGE is run. Actually, this is just a specialized case of the metadata-based conditional activation. The added functionality here is that at SSI&T time PGEs can be grouped into PGE collections. In such cases, the instrument mode would determine which PGE in the collection is run. This mechanism is used primarily to improve the accuracy of plans generated by the PLS.

11.2.3 Error Handling

Error handling is a little different from the other two categories of production rules. This is because it is mostly an operational procedure which occurs in response to an (hopefully) anomalous event. The key mechanism for implementing error handling will be PGE exit codes. Release A PDPS is currently working to define PGE exit codes and determine how best to associate actions with exit codes. (Establishing Science Software Exit Conditions for the Production Environment, 420-WP-006-001) is in preparation. The key enhancement to error handling by Release B is the reuse of the on-demand processing mechanism to automate the response to a specific error code. In this case, when a PGE undergoes SSI&T, On-Demand Production Requests (OPRs) are then associated with various PGE exit codes.

11.2.3.1 Inputs required to implement the Automated Error Handling

At SSI&T time, for every exit condition which will need another PGE to be run, the following information will need to be entered:

Exit Code - PGE exit code which triggers action Message - Message to be displayed to operation console OPR - On-Demand Production Request to be executed. The OPR will have the same timeframe as the original DPR so if the DPR were run on data from 6/20/99 12:00-13:00, then the OPR would be given the same timeframe before being passed to PLS.

11.2.3.2 Example use of Automated Error Handling

It is difficult for a short example to capture the full error handling options of the ECS production system. The following example simply gives a flavor for what is possible with regard to automatically submitting OPRs based on a PGEs exit code. While a certain PGE (PGE1) is undergoing SSI&T it is set up so that if PGE1 has an exit code of 6, it should be re-run in debug mode. If it has an exit code of 12 (can only happen from debug mode), then a second PGE (PGE2) should be run. At a later date, in the operational environment, a DPR for PGE1 is running and terminates with exit code 6. An OPR is generated for PGE1 which includes the runtime flag used to send it into debug mode. A message is displayed on the operator's console informing them of this event. The OPR is run and it finishes with an exit code of 12. Now another OPR is created, this one for PGE2. A different message is displayed on the operator's console.

11.3 Combinations of Production Rules

Introduction

One of the most powerful features of production rules is the ability to combine multiple rules. This gives the science software developers greater flexibility and control over the production of their data. This section is intended to illustrate how some of the production rules can be used together. It is not intended to be exhaustive, but rather to give a sampling of the more common combinations. Most combinations are quite easy to understand and the implications of these combinations are quite clear. However, while theoretically any and all of the production rules can be combined, some combinations will make little sense. For instance, combining tiling with intermittent execution will only produce DPRs for some tiles and it will not be easy for the science software developer to determine those tiles in advance. Conversely, intermittent execution works quite well with the basic temporal rule and behaves in an easily predictable manner. While the software can handle complex combinations, in practice these combinations might have results which are not especially intuitive. For example, while combining alternate inputs with metadata based activation, tiling and intermittent execution would seem quite reasonable to the Planning Subsystem software, it would be difficult for a human to determine the results in advance. However, if intermittent execution is removed from the above combination, the remaining combination might be a perfectly valid production recipe. The following sections provide a few examples of how production rules may be combined.

11.3.1 Basic Temporal

The Basic Temporal rule is fundamental to the production system. All Production Requests will have a temporal component. Consequently, all of the other production rules are being combined with the Basic Temporal rule.

11.3.2 Alternate Inputs and Metadata-Based Conditional Activation

It is possible to combine these two rules so that the metadata of the inputs determines which alternate is used. For example, suppose there is a PGE which, along with its required inputs, can use one of two alternates, but the primary alternate must have a certain level of QA set in its metadata. If the primary data set becomes available, its metadata is checked and, if it fails the check, an attempt is made to use the second alternate.

11.3.3 Alternate Inputs and Data Server Proxy

This combination if quite straightforward. After the input data sets are determined via the Alternate Inputs rule, the Data Server Proxy service is invoked.

11.3.4 Alternate Inputs and Tiling

In this combination determining which data granules fall within a tile is a completely separate activity from determining which alternate is most should be used. In this case the tile definition could be used to acquire the data sets which had be chosen as part of the alternate input process.

11.3.5 Intermittent Execution

While it was mentioned above that all PGEs use the Basic Temporal rule, it is worth emphasizing that point in the case of Intermittent Execution. This rule is somewhat unique since it doesn't create DPRs, it removes them. By definition this rule needs to be used in combination with another rule.

11.3.6 Changes to rules

The following changes have occurred in the listing and organization of the rules documented by this section:

The Release A Basic Temporal rule has been added.

The current Alternate Inputs rule is the consolidation of Optional Inputs, Alternate Inputs - Hierarchical Preference and Alternate Inputs - No Preference. Subsetting and Subsampling have been combined and renamed Data Server Proxy· Alternate Inputs - Temporal/Spatial Tradeoff has been left out. This is because the PDPS team is still determining the best manner to implement this rule. There are several reasons for this including the lack of an IT provided algorithm, the nature of which could impact the design and implementation. Depending on what variables the algorithm uses, this rule might actually be a variant of the Data Server Proxy rule, or it might require the algorithm(s) to be integrated into PLS code.

11.4 Production Rules Technical Information Sources

1 ECS Baseline Information System :

2 PDPS Home Page: http://dmserver.gsfc.nasa.gov/ecsdev/relb/pdps/index.html

http://pete.hitc.com/baseline/ -choose drop4, you will find latest

patch documentation and much more.

3 EOS Instrument Team Science Software (PGE's):

http://ecsinfo.hitc.com/iteams/Science/science.html. Production Rules White Paper and much more.

4 MODIS - Science Data Processing software Release 4 System Description, SDST-104 dated August 25, 1998.

5 Test Scenarios for selected Production Rules can be viewed by accessing the SCF at: **</home/dheroux/DPS/TESTBED/MISR/SSIT/V2/ODL/Scenarios**>

11.4.1 Production_Rules_Syntax

The Production Rules Syntax are presented as part of the Powerpoint Slides that accompany this document.

Production Rules identified thus far are listed as follows:

Basic Temporal, Advanced Temporal, Period Specification, Alternate Inputs, Optional Inputs, Metadata-based Activation, Metadata-Based Query -Static, Metadata-based Query - Dynamic, Orbit-Based Activation, Orbit Path, Runtime Parameter, Multi-File Granules, Multi-Granule ESDT's, Spatial Query, Minimum Number of Granules, Land Tiling, Tiling with Metadata-based Query, Optional DPRs, Ocean Data Day, Most Recent Granule, Alternates based on Minimum number of Granules, Zonal Tiling, Tile Clustering and Smart_Start_of_Year.

11.5 Production Requests

11.5.1 Science Software and Production Requests

Science software is one of the keys to production planning and processing:

- Performs the actual data processing to create desired products.
- Is developed at Science Computing Facilities (SCFs) external to ECS.
- Is embodied in Product Generation Executives (PGEs) when the software is integrated into the ECS production processing environment.
  - PGEs are science software code (e.g., executable programs or shell scripts) that contain the instructions for processing data to create the desired products.

The production request (PR) is another key to production planning and processing. The Production Planner defines ECS science data processing in terms of PRs.

- A PR is an order for data to be produced by the Data Processing Subsystem.
- A single PR may specify several jobs (using the same PGE) that are to be run over a period of time or a single job producing a single set of data.
- PRs may apply to the processing of new data (standard PRs or standing orders) or the reprocessing of existing data (reprocessing PRs).
- Each PR identifies a specific PGE for generating a particular type of product.
  - Some PGEs are dependent on others; i.e., some PGEs require input data that are the output of other PGEs.
  - The planning software will recognize and reject a PR when the PR specifies a PGE that requires data from another PGE that has not yet been specified in a PR.

The Planning Subsystem performs the following functions:

- Uses each PR to generate either one or a series of Data Processing Requests (DPRs).
  - Each DPR corresponds to one execution of a single PGE.
  - Each DPR contains the information that is needed by the SDP processing function, including PGE-related information.
- Checks the availability of the data required for the DPR, either from the data server (if the data have been previously ingested) or from internal predictions (if the data are expected to arrive in the future).
- Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.

Figure 11.5.1-1 shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.

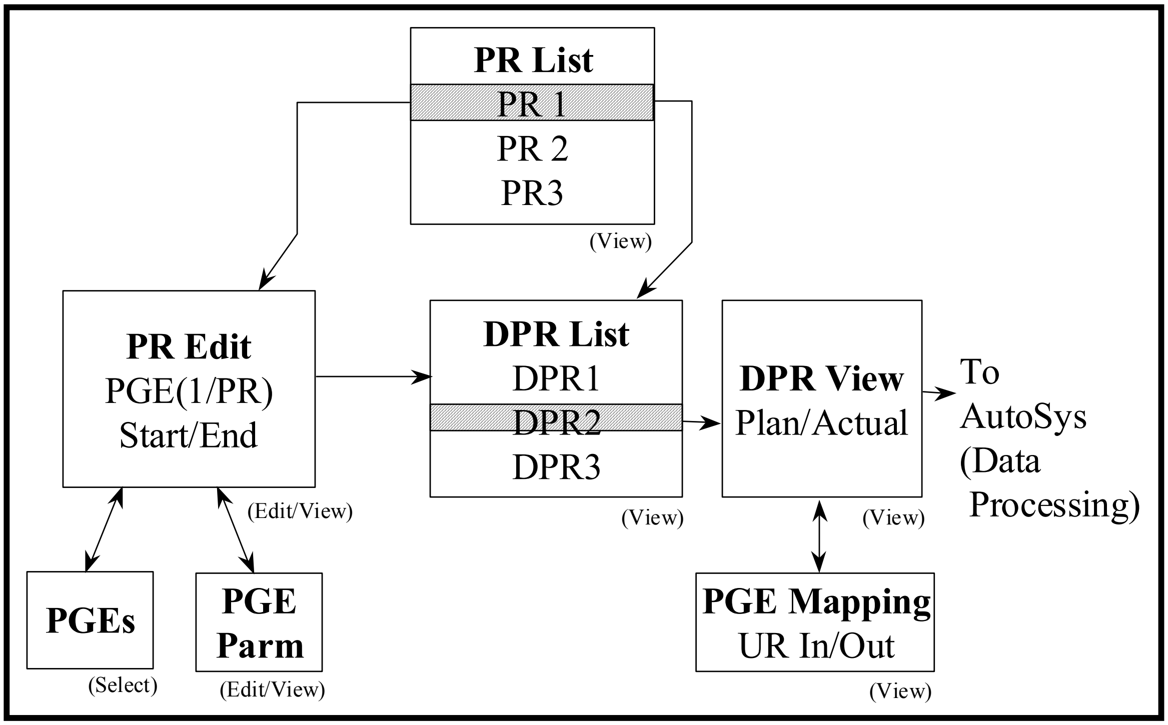


Figure 11.5.1-1. Production Request Editor Flow

11.5.2 Types of Processing

ECS either accommodates or will accommodate the following three general types of data processing:

- Routine Processing
- Reprocessing

- Ad-Hoc Reprocessing
- On-Demand Processing

11.5.2.1 Routine Processing

Routine Processing is pre-defined software production processing that is periodic and keyed to data arrival. For example, every day a Production Planner includes in the daily schedule a DPR for generating a particular Level 1A product from the most recent Level 0 data from the applicable satellite instrument.

11.5.2.2 Reprocessing

Reprocessing typically involves using a new, improved PGE to process data that had previously been processed with an older version of the PGE. In such cases reprocessing would be a large-scale operation, especially if several years worth of data were to be reprocessed. Consequently, the Production Planner is likely to schedule reprocessing in manageable quantities so the processing resources can accommodate routine and on-demand processing in addition to the reprocessing.

11.5.2.3 Ad-hoc Reprocessing

Ad-hoc Reprocessing could be necessary at any time. For example, if a product fails a quality assurance (QA) check, the same PGE could be run again on the same data set in the hope of creating an acceptable product. Similarly, if processing of a PGE fails for some reason, it might be possible to rerun the PGE and hopefully achieve a successful outcome.

11.5.2.4 On-Demand Processing

On-Demand Processing is ad-hoc processing initiated by either the Planning Subsystem or an end-user (as opposed to the Production Planner). For example, a researcher using data from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument on the Terra satellite may need a particular Level 2 product that has not yet been generated. The ASTER researcher would submit an on-demand request to have the product generated from a Level 1B product stored in the archive.

In the future such on-demand processing requests (OPRs) will be entered from a Client Subsystem tool, passed through the Distributed Information Manager (Data Management Subsystem) and the Data Server to the Planning Subsystem. Currently there is a work-around to the automated process which requires the requester to contact DAAC personnel to make the request. So far ASTER researchers are the only identified external users of automated on-demand processing. Another future feature is automated cross-DAAC planning. It is a process that will be undertaken when products produced at one DAAC require inputs being produced at another DAAC. The predicted production time of remote input products will be used in creating local production plans. The primary mechanism for cross-DAAC planning will be the use of Predicted Data Availability Schedules (PDAS), created when a plan is created. A DAAC's PDAS will be made available to remote DAACs via the Data Server.

11.5.2.5 ASTER on Demand Capability

There is some information that needs to be provided to the operations personnel who will be doing the SSIT procedures that create the many PGE Profiles that the ASTER on Demand capability depends on. When ODFRM passes ODMGR the parameters that a user has set for a particular Product they come in a parameter = value format in the GlParameterList. What ODMGR does is attempt to match these parameter/value pairs against Runtime Parameter/Value pairs in the PDPS database that have been entered at SSIT. If the parameter or values do not EXACTLY match what was put in the ODL for the Runtime parameter/value pairs, then ODMGR will not be able to select the correct PGE to create the product.

11.5.2.6 AST_06 Product Guidelines

There are certain parameters that are required for the AST_06 products. However, not all of the parameters have been identified yet.We needed to have 3 different PGEs, each of which produces a different AST_06 product (AST_06VD, AST_06SD, AST_06TD).

For the AST_06 products, the parameters should NOT be set (in the ODL) as selector parameters. The parameters for the AST_06 products are what are called "overwriteable" parameters, meaning that PDPS will overwrite the value in the database with whatever it gets from ODFRM. The parameters that can be set when requesting the AST_06 products are as follows:

MatrixType
SamplingFreq
OutputMean
OutputSigma
BlueBand
GreenBand
RedBand
FirstLine
LastLine
FirstSample
LastSample\

11.5.2.7 Non-Standard L1B on Demand Requests

In support of Non-Standard L1B On Demand Requests, SSI&T personnel at EDC need to add information to the ODL defining the AST_L1B ESDT to PDPS. The installation can set up the database originally but any re-registration of the AST_L1B ESDT (in the PDPS database) will cause the information populated in the database to be deleted.

NCR 25773 has been written against this problem, but unknown is the timeframe for it being fixed. So the ODL file ESDT_AST_L1B#{ESDT version}.odl needs to have the following added for Non-Standard L1B requests:

```
OBJECT = METADATA_DEFINITION
```

```
   CLASS = 1
   PARM_NAME = "InputPointer"
   CONTAINER_NAME = "InputGranule"
   TYPE = "STR"
 END_OBJECT = METADATA_DEFINITION

 OBJECT = METADATA_DEFINITION
   CLASS = 2
   PARM_NAME = "ASTERMapProjection"
   CONTAINER_NAME = "AdditionalAttributes"
   TYPE = "STR"
 END_OBJECT = METADATA_DEFINITION

 OBJECT = METADATA_DEFINITION
   CLASS = 3
   PARM_NAME = "Resampling"
   CONTAINER_NAME = "AdditionalAttributes"
   TYPE = "STR"
 END_OBJECT = METADATA_DEFINITION
```

11.6 Production Rules used in Planning PGE Processing

11.6.1 Production Rules Defining the PGE

Production rules are the instructions about how a particular PGE is to be run. The instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions.

A single PGE may use one or more sets of production rules, known as PGE profiles, since it may be desirable to run the same PGE with different input data sets, or activation conditions. The production rules are entered when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values are entered at that time. The initially selected runtime parameters, metadata check parameters, tile IDs, and many of the default parameters can be overridden in the production environment when a Production Request is entered.

Production rules define the PGE to the Planning and Data Processing Subsystems (PDPS). The following types of conditions can be specified for each PGE:

- The time period for which the PGE will run.
    - A PGE can run every hour, every day, or for every orbit of a satellite. The frequency of how often a PGE runs must be defined to PDPS so that it knows when to plan and execute the PGE. A definition of a satellite's orbit could be included if the PGE were to be executed for some number of satellite passes.
- PGE Inputs.

  - A PGE can have any number of inputs. The types of the inputs and how frequently they are available helps determine on what basis the PGE is scheduled.
  - Most inputs to a PGE are retrieved based on time; the specified inputs are retrieved from the Data Server Subsystem for the time which the PGE is defined to execute. Production Rules allow other conditions to be added to the mix, such as checks or queries against the metadata of the input granules, or the lists of inputs as alternates (for when a primary input is not available) or optionals (for inputs without which the PGE can still run successfully). If inputs are defined as alternate or optional, the number of inputs staged for the execution of the PGE may vary from one run to the next.
- PGE Outputs.
  - A PGE can have any number of outputs. The characteristics of the outputs can have effects on any downstream PGEs that use them as inputs. For example, it is possible for an output to be defined as optional, in which case it may or may not even be produced. (When an output is not generated, it cannot be used as input for a downstream PGE.)
- Runtime Parameter Values.
  - A PGE can have any number of runtime parameters, which are values that are placed in the process control file (PCF) under specified logical IDs before the PGE executes. The PGE treats them as constants and normally they are set either during SSI&T or when the Production Request is entered.
  - For some production rules (such as Orbital Processing) there is specific information that can be placed in a runtime parameter if so desired by the PGE.
- Geographic Tiles.
  - A PGE can define a geographic location for which it will process data. Tiles are defined through production rules, and change the staging of inputs from time-based, to a combination of time- and geographically-based. Data are retrieved based on their location on the Earth with respect to the tile that it is currently being processed.

Some (but not all) production rules can work with other production rules.

11.6.1.1 Production rules are often used for the selection of dynamic inputs.

- **Dynamic inputs** can be either internal or external.
  - **Dynamic internal** inputs are produced by other PGEs (they are called dynamic internal inputs because they are produced at an ECS DAAC).
  - **Dynamic external** inputs are periodically ingested and stored in the Data Server Subsystem (they are termed dynamic external inputs because they are produced outside of the DAAC).

- **Static inputs** are granules that are inserted during the SSI&T process and are retrieved not on the basis of time but by Earth Science Data Type (ESDT) and science group.
  - The Metadata Query Production Rule is the only production rule that works for choosing static inputs.

PGE profiles allow a PGE to be defined to PDPS multiple times, each with a different set of inputs, outputs, or even scheduling information. Each PGE's definition is made up of its name, its version and its profile number. Different PGE name/version pairs define different PGEs to PDPS. The addition of the profile allows for multiple definitions of a PGE name/version pair. There can be up to 99 profiles for each PGE.

11.6.1.2 Syntax of Production Rules

Production rules are defined in the following two ways:

- Through science metadata that is entered in various types of files during the SSI&T process.
- By entering parameter values when a Production Request is created to schedule the PGE.

During SSI&T, production rules are defined in files written in Object Description Language (ODL) in a parameter equals value format. There are three general categories of ODL files:

- PGE Science Metadata ODL Files.
- ESDT Science Metadata ODL Files.
- Production Rule-Specific Science Metadata ODL Files
  - Orbit Definition ODL Files.
  - Path Map Definition ODL Files.
  - Tile Definition ODL Files.

When a Production Request is created to schedule a PGE, it is necessary to enter certain information that is essential to implementing the production rules that affect the particular PGE. The information may concern the date and time-range

11.6.1.3 PGE Science Metadata ODL Files

The PGE science metadata ODL file defines a PGE (or at least the current plan for its operation) to PDPS. It specifies everything from the PGE name and version, to the period for the PGE (how often it runs), all inputs and outputs, any runtime parameters and any exit messages or dependencies. A template version of the PGE science metadata ODL file is created by the **SSIT Create ODL Template** program from a PCF from the PGE.

11.6.1.4 ESDT Science Metadata ODL Files

The ESDT science metadata ODL file defines a PGE input or output to PDPS. Each input and output of a PGE must have a corresponding ESDT science metadata ODL file defined. It describes everything that PDPS needs to know about the subject input or output file, from its name and version, to its period (how often data is collected), to where it is used and archived. Note that many PGEs can use the same input or output type, and thus can share the same ESDT science metadata ODL file.

Unlike the PGE science metadata ODL file, there is no tool for automatically generating a template ESDT science metadata ODL file. A template version exists under the data directory called ESDT_ODL.template. The template must be copied to a file that follows the naming convention *ESDTShortName#ESDTVersionID*.odl.

11.6.1.5 Production Rule-Specific Science Metadata ODL Files

The production rule-specific science metadata ODL files provide specific information to PDPS about production rules used by a PGE. They are needed only when the PGE is subject to one of the following conditions:

- Is executed on the basis of a satellite orbit.
- Needs to know the orbital path of a satellite.
- Requires data based on geographic tiling of the Earth.

Since not every PGE is based on orbits or tiles, not all PGEs require these files. The comments in the PGE_ODL.template describe when setting a specific parameter means that a production rule-specific science metadata ODL file needs to be created.

The production rule-specific science metadata ODL files are broken into three types, which are defined as follows:

- Orbit ODL File.
  - Defines the orbital period of the satellite from which the PGE's input data is created.
  - Defines when a given orbit starts, how long it lasts, and the number of the orbit.
  - PDPS uses the information in the orbit ODL file to extrapolate future orbits and is able to plan PGEs that are required to run every so many orbits of the satellite.
- Pathmap ODL File.
  - Defines the mapping between the cyclic 0-233 orbits that the satellite makes with the actual path number that the PGE requires.
  - PDPS computes the path number from the orbit number (specified in the orbit ODL file) by incrementing it until it reaches the 233 maximum, then resetting it to zero.
  - Many instruments expect the path number to be a fixed swath on the Earth, so it is not just incremented for each satellite pass.
  - The pathmap ODL file creates a mapping from the sequential path numbers to the path numbers expected by the PGEs.

- Tile ODL File.
  - Defines the coordinates of the tiles used by some instruments to specify geographic locations on the Earth.
  - The tile definitions are used by PDPS to schedule the PGE (one execution per tile) and to acquire the necessary data (using the geographic coordinates to acquire data for the tile being processed only).

Unlike the PGE science metadata ODL file, there is no tool to automatically generate a template production rule-specific science metadata ODL file. Because the files themselves tend to be small, this is not usually a problem. A template version of each kind of production rule-specific science metadata ODL file (e.g., ORBIT_ODL.template, TILE_ODL.template) exists in the /usr/ecs/<MODE>/CUSTOM/data directory on the AIT Workstation. The templates must be copied, named properly, and edited in order to create the appropriate production rule-specific science metadata ODL file.

11.7 Release 5 Production Rules

The following statements provide some simplified descriptions of production rules made available in Release 5:

- **Basic Temporal** - Temporal (time) range of inputs matches the temporal range of outputs.
- **Advanced Temporal** - Temporal range of inputs is offset from the expected temporal range of inputs and outputs.
- **Alternate Input** - PGE is run with different inputs based on the availability of various alternate input data sets.
- **Optional Input** - PGE is run with specified optional inputs if available; otherwise, PGE is run without them.
- **Minimum/Maximum Number of Granules** - Minimum number of input granules needed for full data coverage and maximum number of input granules to search for may be specified. Minimum and maximum number of outputs expected from the PGE may be specified.
- **Optional DPRs** – The only DPRs executed are those for which the non-routine key input data actually become available (i.e., are either produced in data processing or can be acquired from the archive).
- **Intermittent Activation** - Every *n*th DPR is activated; all other DPRs are skipped.
- **Metadata Checks** - DPR is run only if input data’s metadata value(s) meet(s) certain criteria.
- **Metadata Query** - Input granule selection is based on metadata value.
- **Data Day** - Input data selection is based on Data Day.
- **Spatial Query** - Input granule selection is based on the spatial coverage of another input (i.e., the key input).
- **Tiling** - Input data is chosen on the basis of Instrument Team-defined tiles (geographic areas).

- **Closest Granule** – DPR is generated if a required input granule within a particular time range (rather than an exact time) is available; otherwise, no DPR is generated. (Supersedes the Most Recent Granule Production Rule)
- **Orbital Processing** - Selection of input times is based on orbit information.

11.7.1 Basic Temporal Production Rule

The Basic Temporal Production Rule defines the timeframe for the PGE along with its input and output data. PGEs subject to the Basic Temporal Production Rule generally have the following characteristics in common:

- Typically scheduled to run using input data that become available periodically (every hour, every day, etc.).
- Use input data for a particular period of time.
- Produce output for a specified length of time.

The data the PGE takes in (its input) and the data it produces (its output) have the same period (or some subset of the same period) as the PGE.

- Example One:
  - A MODIS PGE processes data for five-minute intervals, producing Level 1B granules.
  - The PGE requires as input the specific five-minute Level 1A granule that is contemporaneous with (covers the same five-minute time period as) the Level 1B granule to be produced.
  - Using the Basic Temporal Production Rule, a five-minute Level 1A granule is staged as input to the PGE and a five-minute Level 1B granule is expected as output, both matching the timeframe for which the PGE is run.
- Example Two:
  - A CERES PGE processes data for 24-hour intervals, producing 24-hour Level 1A granules as output.
  - As input the PGE takes Level 0 data that is ingested every two hours.
  - Using the Basic Temporal Production Rule, twelve two-hour Level 0 granules are staged as input to the PGE and a 24-hour Level 1A granule is expected as output, matching the timeframe for which the PGE is run.

The fundamental elements used to define the Basic Temporal Production Rule are “period: and “boundary.”

- **Period** is the length of time for which a PGE processes data or the length of time for which input and output data is collected.
  - A PGE that is subject to the Basic Temporal Production Rule only and that processes data in two-hour blocks, takes in data that relates to a particular two-hour interval and produces output data for that same two-hour period.

  - Data that has a period of 15 minutes was collected or produced for a 15-minute time period.
- **Boundary** is the starting point for the data or PGE. Depending on the characteristics of the data or PGE, the boundary may be the start of a minute or hour or day or week (etc.).
  - If a PGE's boundary is the start of the hour, it processes data that starts every hour and runs on data for the length of its period.
  - If data comes in every day, PDPS predicts that the data is going to be available at the start of the day and allows scheduling of PGEs that use the data as input accordingly.

Both the PGE itself and the input data have a boundary and period associated with them. That is how PDPS determines the frequency of processing for a Basic Temporal PGE and the time period for its inputs and outputs.

PDPS uses **period** and **boundary** in combination to plan the processing of each PGE, including determining its input requirements and anticipated output (which may be input to other PGEs). If a PGE has a period of one hour and a boundary of “start of day,” it is scheduled every hour, beginning at midnight. If an input has a period of 15 minutes and boundary of “start of hour,” PDPS predicts it every 15 minutes beginning on the hour.

Boundary offset is an addition to the Basic Temporal Production Rule that allows a PGE or data to start on an offset from a given boundary. For example, if a PGE would normally run every day but not start until two or three hours into the day (e.g., beginning at 3:00 a.m. instead of midnight), a boundary offset can be used to add three hours to the “start of day” boundary. This would mean the PGE would run on data that occurred three hours after the boundary.

Data with offset times refers to data where the start time is a few minutes off of the start time that PDPS expects. For example: if data is defined to PDPS as follows:

```
BOUNDARY = "START_OF_HOUR"

PERIOD   = "HOURS=1"
```

but the data actually starts at 1:05 and ends at 2:05, the data is said to have **offset times**. There is a flag in the production rule syntax that tells PDPS to shift granule time specifications to match the granules in the archive.

The **end-of-month anomaly** is an addition to the Basic Temporal Production Rule that allows a PGE or data to cover a specific number of days within a month. The month is broken into thirds. The first third is composed of the first 10 days of the month. The second third consists of days 11 through 20. And the last third varies in length depending on the total number of days in the month (i.e., for November it would have 10 days; for December it would have 11 days). A specific **boundary** and **period** allow a PGE or its data to be scheduled into thirds of a month.

Figure 11.7.1-1 provides an illustration of the Basic Temporal Production Rule. The PGE has a boundary of “start of day” and a period of one hour, so it is scheduled for every hour through the day. If a Production Request were entered for two full days of processing, a DPR would be created for the PGE to run every hour; i.e., 48 DPRs total. If a Production Request were created for a four-hour period in the middle of a single day (for example, from 12:00 noon to 4:00 p.m.), then four DPRs would be created, one for 12:00-1:00, one for 1:00-2:00, one for 2:00-3:00, and one for 3:00-4:00.

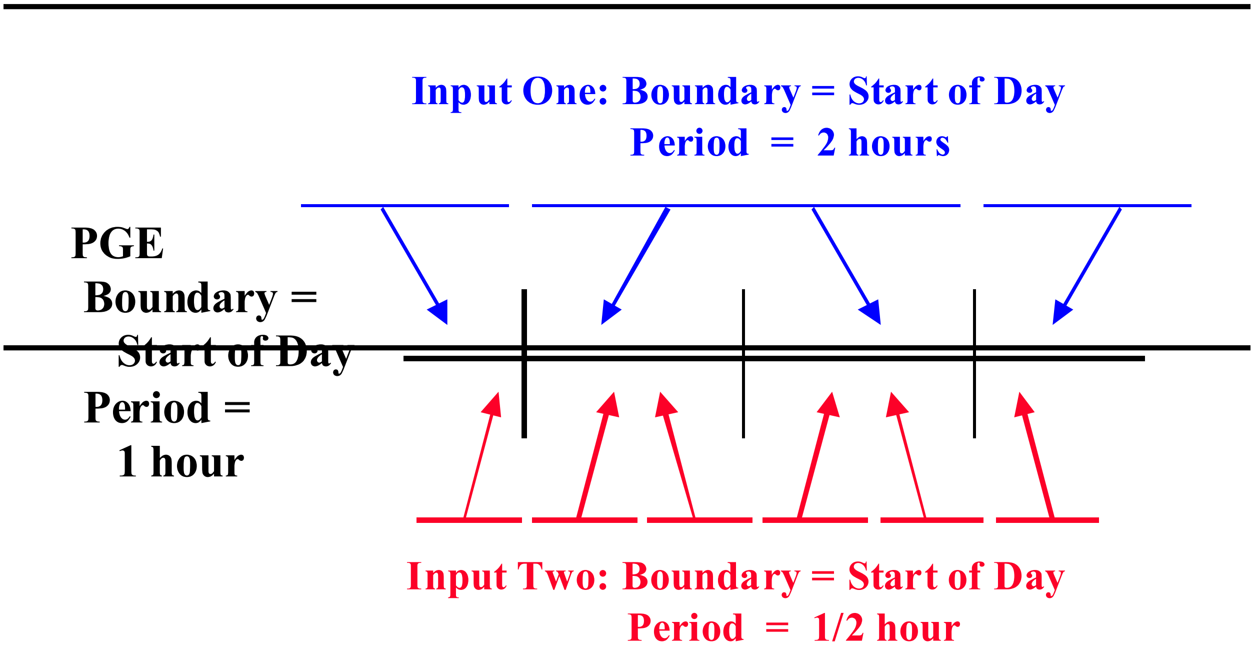


Figure 11.7.1-1. Example of the Basic Temporal Production Rule

In the example (Figure 11.7.1-1), Input One has a boundary of "start of day" and a period of two hours, so when PDPS plans for its availability, it expects a granule every two hours beginning at midnight. Consequently, each granule of Input One is associated with two DPRs for the PGE, because the PGE encompasses only one hour of the two-hour granule's period.

Input Two has a boundary of "start of day" and a period of ½ hour, so when PDPS plans for its availability, it expects a granule every ½ hour beginning at midnight. As a result two granules of Input Two are associated with each DPR for the PGE, because the PGE encompasses an hour of the ½-hour granule's Period. Thus, every DPR of the PGE will wait for two granules of Input Two to arrive before it can be processed.

11.7.1.1 PGE Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable PGE science metadata ODL file in order to implement the Basic Temporal Production Rule:

- SCHEDULE_TYPE.

- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The SCHEDULE_TYPE parameter specifies the type of scheduling that will be done for the PGE. The following values are applicable to the Basic Temporal Production Rule:

- "Time"
  - The PGE is scheduled on the basis of the specified boundary/period and the availability of data for that boundary/period.
- "Snapshot"
  - The PGE is scheduled for a single date/time.
  - Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY are **not** needed when "Snapshot" is specified.

Other values for SCHEDULE_TYPE apply to other production rules, such as the following values:

- “Data"
  - The PGE is scheduled on the basis of the availability of data produced by other PGEs.
- "Tile"
  - The PGE is scheduled based on the definition of geographic tiles.
- "Orbit"
  - PGE scheduling is based on the orbit of the spacecraft.

The PROCESSING_PERIOD parameter describes the length of time for which the PGE executes. Data will be acquired (barring any combination of Production Rules) for the specified period and output data will be planned for the given period. It is of the format "<Period Type>=<Length of Period>". Note that “length of period” can be specified as a positive integer only. The following values are acceptable “period type” entries for the Basic Temporal Production Rule:

- "YEARS"
  - PGE processes data applicable to a given year or years.
  - "YEARS" might be specified for a PGE that computes a yearly average.
  - For example, PROCESSING_PERIOD = "YEARS=1" relates to a PGE that processes one year’s worth of data.
- "MONTHS"
  - PGE processes data applicable to a particular month or several months.
  - “MONTHS” is most likely to be used for some kind of averaging PGE.
  - For example, PROCESSING_PERIOD = "MONTHS=2" relates to a PGE that processes two months’ worth of data at a time.
- "THIRDS"

  - PGE processes data applicable to some number of thirds of the month.
  - For example, PROCESSING_PERIOD = "THIRDS=1" relates to a PGE that processes data applicable to 1/3 of the month.
- "WEEKS"
  - PGE processes data applicable to some number of weeks.
  - For example, PROCESSING_PERIOD = "WEEKS=2" relates to a PGE that processes two weeks' worth of data every time it runs.
- "DAYS"
  - PGE processes data applicable to some number of days.
  - For example, PROCESSING_PERIOD = "DAYS=5" relates to a PGE that processes five days' worth of data.
- "HOURS"
  - PGE processes data applicable to some number of hours.
  - For example, PROCESSING_PERIOD = "HOURS=4" relates to a PGE that processes four hours' worth of data when it is executed.
- "MINS"
  - PGE processes data applicable to some number of minutes.
  - For example, PROCESSING_PERIOD = "MINS=5" relates to a PGE that processes five minutes' worth of data.
- "SECS"
  - PGE processes data applicable to some number of seconds.
  - For example, PROCESSING_PERIOD = "SECS=2" relates to a PGE that runs on two seconds' worth of data.

There are other types of values for PROCESSING_PERIOD but they apply to other production rules (as described in the applicable sections of the lesson).

The PROCESSING_BOUNDARY parameter specifies the boundary (starting point in time) of the PGE. It tells when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction to schedule the PGE.

The following PROCESSING_BOUNDARY values are used for implementing the Basic Temporal Production Rule:

- "START_OF_HOUR" – PGE processes data for each hourly interval.
- "START_OF_6HOUR" - PGE processes data for every 6-hour interval.
- "START_OF_DAY" - PGE processes data for every daily interval.
- "START_OF_WEEK" - PGE processes data for every weekly interval.
- "START_OF_ONE_THIRD_MONTH" - PGE processes data for every 1/3 of a month.

- "START_OF_MONTH" - PGE processes data for every monthly interval.
- "START_OF_YEAR" - PGE processes data for every yearly interval.
- "START_DATE=*DD/MM/YYYY*" - PGE processes data for the specified date only.

There are other values for PROCESSING_BOUNDARY that apply to other production rules (as described in the applicable sections of the lesson).

11.7.1.2 Handling Data with Offset Times

When the ALIGN_DPR_TIME_WITH_INPUT_TIME flag is set to "Y" (i.e., ALIGN_DPR_TIME_WITH_INPUT_TIME = "Y") PDPS shifts the expected times for input data to the actual times found in the archive. If the flag is NOT set, data with offset times can cause problems when generating Production Requests.

11.7.1.3 ESDT Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable ESDT science metadata ODL file in order to implement the Basic Temporal Production Rule:

- DYNAMIC_FLAG.
- PERIOD.
- BOUNDARY.

The DYNAMIC_FLAG describes the type of data that is defined in the ESDT science metadata ODL file. It specifies to PDPS what kind of data the PGE requires as input or produces as output. It can have any of the following four possible values, all of which are valid for Basic Temporal data:

- "S"
  - Static Data.
  - Data do not change at regular intervals.
  - The same granule can be used as input for many runs of the PGE.
  - Calibration files are a good example of static data.
- "I"
  - Dynamic Internal.
  - Data are produced by a PGE running at the local DAAC.
  - All output products are either "dynamic internal" or "interim" kinds of data.
- "E"
  - Dynamic External.
  - Data are produced by an external source (not a PGE running at the local DAAC).
  - EDOS data is a primary example.
  - Dynamic external can be set for PGE inputs only.

- "T"
  - Interim/Intermediate.
  - Data are stored only temporarily by the Data Server Subsystem.

The PERIOD parameter specifies the length of time covered by the data. Data are expected to be either ingested or produced for the length of the PROCESSING_PERIOD described in PGE science metadata ODL files. However, the PERIOD of the data does **not** have to match the PROCESSING_PERIOD defined for the PGE. PDPS plans for data where the ESDT period is less or more than the processing period of the PGE that uses it. For example, if the PGE PROCESSING_PERIOD = "HOURS=1" and the input data PERIOD = "MINS=5", then PDPS plans to acquire twelve granules of the input data to cover the PROCESSING_PERIOD.

The following "period type" values are used for implementing the Basic Temporal Production Rule:

- "YEARS"
  - Data span a year or years.
  - "YEARS" might be selected for a yearly average output product.
  - For example, PERIOD = "YEARS=1" specifies data that cover a period of a year.
- "MONTHS"
  - Data span a month or several months.
  - "MONTHS" is most likely used for some kind of averaging output product.
  - For example, PERIOD = "MONTHS=2" specifies data that cover a period of two months.
- "THIRDS"
  - Data span some number of thirds of a month.
  - For example, PERIOD = "THIRDS=1" specifies data that cover a period of 1/3 month.
- "WEEKS"
  - Data span some number of weeks.
  - For example, PERIOD = "WEEKS=2" specifies data that cover a period of two weeks.
- "DAYS"
  - Data span some number of days.
  - For example, PERIOD = "DAYS=5" specifies data that cover a period of five days.
- "HOURS"
  - Data span some number of hours.

  - For example, PERIOD = "HOURS=4" specifies data that cover a period of four hours.
- "MINS"
  - Data span some number of minutes.
  - For example, PERIOD = "MINS=5" specifies data that cover a period of five minutes.
- "SECS"
  - Data span some number of seconds.
  - For example, PERIOD = "SECS=2" specifies data that cover a period of two seconds.
- "ORBITS"
  - Data span some number of orbits of the spacecraft.
  - For example, PERIOD = "ORBITS=1" specifies data that cover one orbit.
  - A PGE can be time-scheduled (using the Basic Temporal Production Rule) but use orbit-based data.

The BOUNDARY parameter is the starting point in time of the data granule. It tells when each data granule should start. Note that the BOUNDARY and PERIOD are used in conjunction to determine the starting and ending time for the granules.

The following values for BOUNDARY apply to the Basic Temporal Production Rule:

- "START_OF_HOUR"
  - Data granules start every hour.
- "START_OF_6HOUR"
  - Data granules start every six hours.
- "START_OF_DAY"
  - Data granules start every day.
- "START_OF_WEEK"
  - Data granules start every week.
- "START_OF_ONE_THIRD_MONTH"
  - Data granules start every 1/3 of a month.
- "START_OF_MONTH"
  - Data granules start every month.
- "START_OF_YEAR"
  - Data granules start every year.
- "START_OF_ORBIT"
  - Data granules start every orbit.

11.7.2 Advanced Temporal Production Rule

The Advanced Temporal Production Rule allows for input data to be acquired for a time period other than that of the PGE or its planned inputs/outputs. It provides an offset mechanism, specifying on an input basis that the data required for processing is some number of seconds earlier or later than the planned time period for the PGE.

- Example One:
  - A PGE requires data from its previous execution for interpolation purposes (e.g., one of its inputs is the output of the very same PGE the last time that it ran).
  - If the PGE processes data for each one-hour interval (producing an hourly product), the Advanced Temporal Production Rule is specified with an offset of minus 3600 seconds (one hour) for the input of the ESDT produced by previous runs.
- Example Two:
  - A PGE takes as input two-hour Level 0 data to produce an L1A product.
  - Because the edges of the Level 0 data can be difficult to process without preceding and succeeding data, the PGE requires three Level 0 granules, one from the time period before it runs, one for the time period it is currently processing and one for the next time period.
  - The PGE is defined as having three inputs, the first with and Advanced Temporal offset of minus 7200 seconds (two hours), the second with no Advanced Temporal offset and the third with an Advanced Temporal offset of plus 7200 seconds (two hours).

The Advanced Temporal Production Rule uses the times specified in the Basic Temporal Production Rule as a reference point for specifying offset(s) to request data from a “period” and/or “boundary” different from that of the DPR or its input. The offsets are specified as either negative or positive numbers to indicate whether the time period of the input data is before or after that of the DPR (a particular run of a PGE).

- **Begin Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR start time. A negative beginning offset requests data that was collected before the DPR start time. A positive beginning offset requests data with a collection time after the start time of the DPR.
- **End Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR end time. A negative ending offset requests data that ended collection before the DPR end time was reached. A positive ending offset requests data that ended collection after the end time of the DPR boundaries.

Note that the beginning and ending offsets are not absolute cut-offs for data. Overlapping granules (granules that start or end outside of the offsets) will be staged as inputs to the DPR.

Figure 11.7.2-1 provides an illustration of the Advanced Temporal Production Rule. The PGE shown in the example processes data for every one-hour interval. However, Input One comes in at two-hour intervals and Input Two is produced every 1/2 hour.

Both the Begin Period Offset and End Period Offset for Input One are -7200 seconds (minus two hours). Consequently, every DPR will stage the "previous" Input One. This could be used to get the "previous" or "next" granule of an input.

The Begin Period Offset for Input Two is zero, meaning that it will match the Start Time of the DPR. The End Period Offset is +1800 seconds (plus 1/2 hour). Therefore, all Input Two granules will be staged that fall within the time period of the DPR plus 1/2 hour. The effect is to acquire all Input Two granules within the time period of the DPR, plus the one from the next 1/2-hour time period, for a total of three granules. The additional granule acquired by means of the End Period Offset might be used for interpolation purposes at the end point.

The same types of parameter settings that apply to the Basic Temporal Production Rule apply to the Advanced Temporal Production Rule. In addition, there are some parameters in the PGE science metadata ODL file that apply specifically to the Advanced Temporal Production Rule. However, the values applicable to the Basic Temporal Production Rule must be set before the Advanced Temporal Production Rule syntax is added.

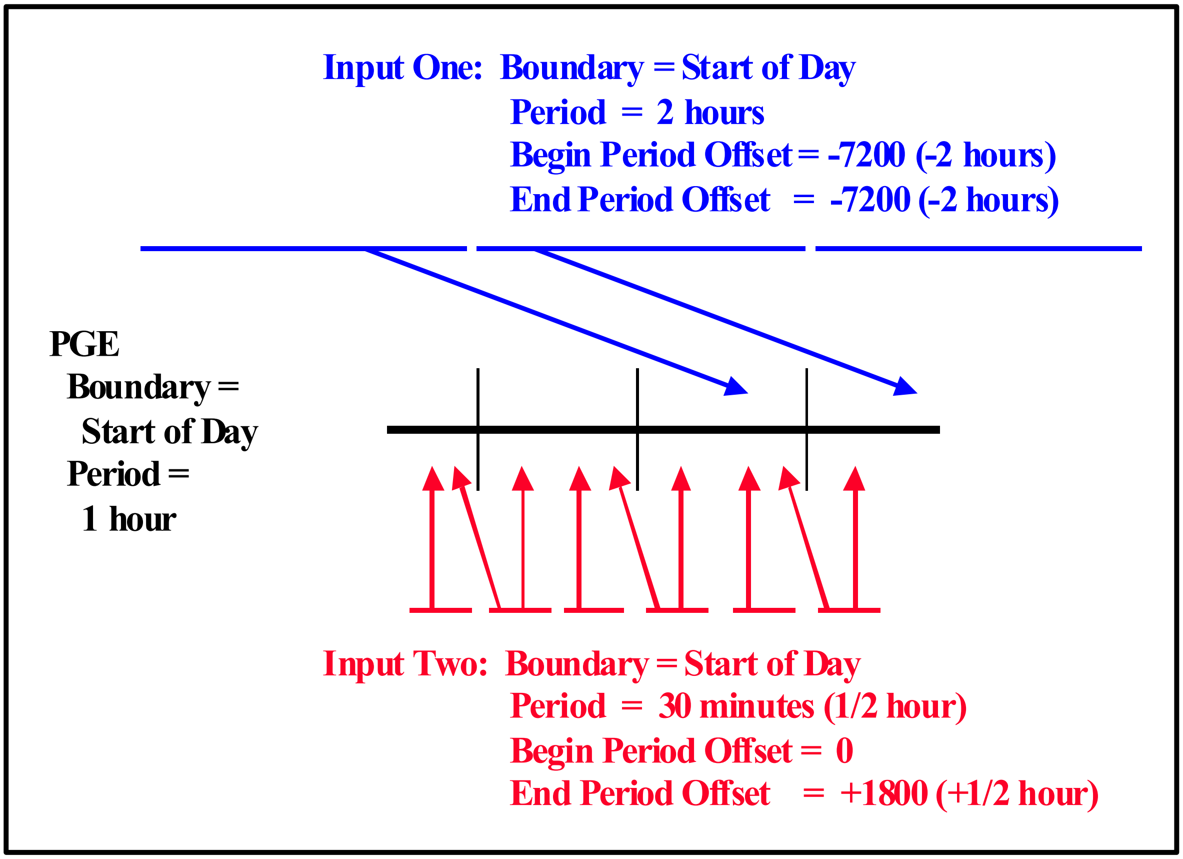

Input One: Boundary = Start of Day
Period = 2 hours
Begin Period Offset = -7200 (-2 hours)
End Period Offset = -7200 (-2 hours)
PGE
Boundary =
Start of Day
Period =
1 hour
Input Two: Boundary = Start of Day
Period = 30 minutes (1/2 hour)
Begin Period Offset = 0
End Period Offset = +1800 (+1/2 hour)

Figure 11.7.2-1. Example of the Advanced Temporal Production Rule

11.7.2.1 PGE Science Metadata ODL File Parameters

During the SSI&T process the PGE science metadata ODL file is generated from the PCF delivered with the science algorithm. A PCF_ENTRY object is generated for each file entry in the PCF. In order to implement the Advanced Temporal Production Rule the PCF_ENTRY object for each type of input file to which the rule applies uses the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   BEGIN_PERIOD_OFFSET =
   END_PERIOD_OFFSET =
   .
   .
   .
END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Advanced Temporal Production Rule:

- BEGIN_PERIOD_OFFSET.
- END_PERIOD_OFFSET.

BEGIN_PERIOD_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR. The value assigned to BEGIN_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection Start Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection Start Time (looking backward in time). For example, BEGIN_PERIOD_OFFSET = -3600 requests data that was collected one hour (3600 seconds) before the DPR start time.

END_PERIOD_OFFSET is the offset added to or subtracted from the Data Collection End Time of the DPR. The value assigned to END_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection End Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection End Time (looking backward in time). For example, END_PERIOD_OFFSET = +2700 requests data that was collected 45 minutes (2700 seconds) after the DPR end time.

The BEGIN_PERIOD_OFFSET and END_PERIOD_OFFSET parameters can be specified for any input PCF_ENTRY in the PGE science metadata ODL file. If not specified, the parameters are set to zero (0) and the Advanced Temporal Production Rule does not apply to the PGE.

11.7.3 Alternate Input and Optional Input Production Rules

The Alternate Input and Optional Input Production Rules are very similar and use much the same processing in PDPS. Both rules allow a PGE to select various inputs based on timers and priority lists. The major difference is that Alternate Inputs requires that one of alternates on the list be used, whereas Optional Inputs allows successful execution of the PGE if no optional input on the list is available.

The Alternate Input Production Rule allows for a PGE to evaluate a list of inputs in priority order and be scheduled and executed with the best priority input that could be found. In essence, a PGE using Alternate Inputs is saying "I would like to run with Input A, but if it's not available, I am willing to run with Input B." A timer can be used to specify how long to wait for a given alternate choice before proceeding with a choice of lesser priority. The PGE is not executed until one of the alternate choices has been found.

- Example:
  - A PGE requires model wind data as an input but is capable of accepting wind data from a Data Assimilation Office (DAO) model, a National Centers for Environmental Prediction (NCEP) model, or (as a last resort) climatology.
  - The PGE would use the Alternate Input Production Rule to list each input in priority order, giving a timer value for how long to wait before trying the next input.
  - If the DAO data are most desirable, DAO would be listed as first choice or "primary" data.
  - NCEP would be the second choice.
  - Climatology would be the last choice.
  - If a timer value is specified for DAO data, the PGE will wait for that timer to expire before running with either NCEP data or climatology.
  - If a timer had been placed on the NCEP input, the PGE would wait before running with the climatology data.

11.7.3.1 The Optional Input Production Rule

Allows for a PGE to list inputs that are desired but not required for it to execute. The inputs are ranked as previously stated and timers are set to wait before choosing a lower-priority type of input. However, if none of the inputs on the list becomes available, the PGE starts because the alternatives are classified as "optional." In essence the PGE is saying "I would like to run with Input A, but if its not available, I can run (and produce reasonable output) without it."

- Example:
  - It would be preferable to run a particular MODIS PGE with the output of a MISR PGE as input.
  - However, the MISR output may not be produced every day.
  - So the MODIS PGE lists the MISR input as optional with a two-hour timer.
  - On those occasions when no MISR output is produced, the MODIS PGE waits for two hours and then is executed without the MISR input.

Figure 11.7.3-1 provides an illustration of the Alternate Input Production Rule. The PGE in the illustration has two inputs that are "required" so they must available for the PGE to be run. It also has one input that is "alternate." The alternate input can be one of three choices, the first choice is the **primary**, then there are second and third choices.

After the pair of required inputs has become available, the alternate inputs are evaluated as follows:

- If the primary alternate is available, it is used as input and the PGE is scheduled for execution.
- There is a one-hour timer on the primary alternate. If the primary alternate is unavailable, the PGE waits until the primary alternate becomes available or the one-hour timer expires, whichever occurs first.
- If the second alternate is available after the timer for the primary alternate has expired, the second alternate is used as input and the PGE is scheduled for execution.

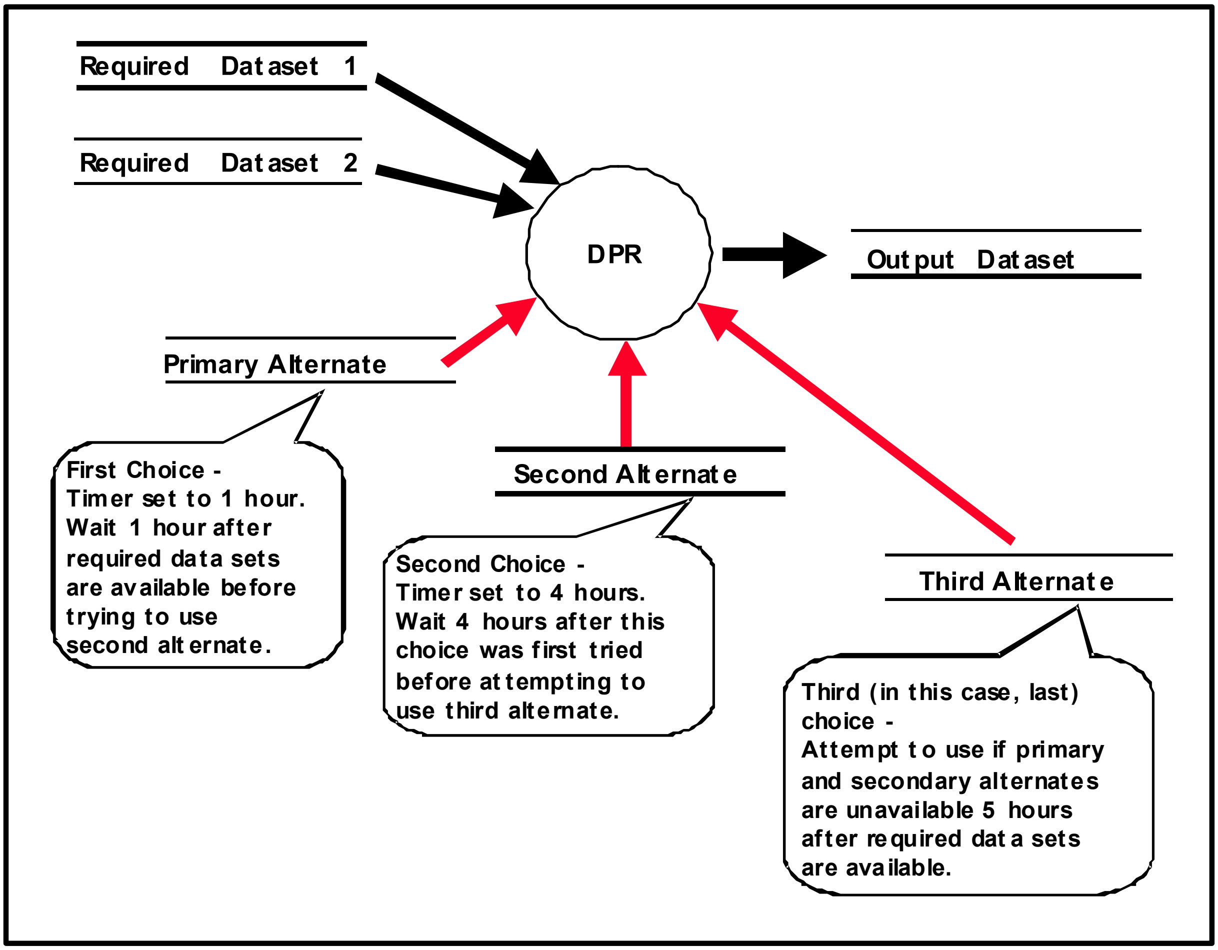

Required Dataset 1
Required Dataset 2
DPR
Output Dataset
Primary Alternate
First Choice - Timer set to 1 hour. Wait 1 hour after required data sets are available before trying to use second alternate.
Second Alternate
Second Choice - Timer set to 4 hours. Wait 4 hours after this choice was first tried before attempting to use third alternate.
Third Alternate
Third (in this case, last) choice - Attempt to use if primary and secondary alternates are unavailable 5 hours after required data sets are available.

Figure 11.7.3-1. Example of the Alternate Input Production Rule

- There is a four-hour timer on the second alternate. If the second alternate is unavailable, the PGE waits until either the primary alternate or the secondary alternate becomes available or the four-hour timer expires, whichever occurs first.
- If the third alternate is available after the timer for the second alternate has expired, the third alternate is used and the PGE is scheduled for execution.
- There is no timer on the third alternate. If the third alternate is not available, the PGE waits until either the primary alternate, the secondary alternate, or the third alternate becomes available, whichever occurs first.
- The PGE will not start processing until one of the alternates becomes available.

If instead of an alternate the third input for the PGE had been defined as an optional input, the preceding scenario would have been the same, except that if neither the primary alternate, the second alternate nor the third option was available after the timers had expired, the PGE would not wait; it would be scheduled for execution without the third input. It would run with the two required inputs only.

The Alternate Input and Optional Input Production Rules are additions to settings/syntax put into the ODL files for other production rules. Inputs deemed “optional” or “alternate” can be searched for and acquired by other production rules (e.g., Basic Temporal or Metadata Checks/Query). The syntax for the rules used to search for the inputs have to be filled out in addition to the syntax required to make the input an alternate or optional input.

11.7.3.2 PGE Science Metadata ODL File Parameters

The following parameter must be set properly in the applicable PGE science metadata ODL file in order to implement the Alternate Input or Optional Input Production Rule:

- INPUT_TYPE.

In addition, one of the following two ODL objects is used within a PCF_ENTRY to define either the Alternate Input Production Rule or the Optional Input Production Rule:

- ALTERNATE_INPUT object.
- OPTIONAL_INPUT object.

INPUT_TYPE is a type of data defined by a PCF_ENTRY object (i.e., between OBJECT = PCF_ENTRY and END_OBJECT = PCF_ENTRY). It can have one of four possible values, only three of which are used to define an alternate or optional inputs:

- "Required"
  - A required input.
  - The data must be available or the PGE does not execute.
  - It is the "normal" value for the parameter (i.e., INPUT_TYPE = “Required”); consequently, the input is neither an alternate input nor an optional input.

- "Primary"
  - The primary alternate input.
  - The data is the first choice in a list of alternates.
- "Alternate"
  - An alternate input (except the primary alternate) in a list of alternates.
  - The data is not the first choice in a list of alternates; it is a subsequent choice if the primary (or a higher-priority alternate) is not available.
- "Optional"
  - An optional input.
  - Availability of the data will be checked and if a timer has been specified, execution of the PGE will wait.
  - The PGE can be executed without the data if it is not available.

Although the Alternate Input and Optional Input Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY; i.e., the ALTERNATE_INPUT object and the OPTIONAL_INPUT object.

The ALTERNATE_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = ALTERNATE_INPUT
      .
      .
      .
   END_OBJECT = ALTERNATE_INPUT
END_OBJECT = PCF_ENTRY
```

The ALTERNATE_INPUT ODL object surrounds an Alternate Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Alternate Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an ALTERNATE_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- WAITFOR.

- TEMPORAL [not implemented].

CLASS is a simple counter used to differentiate the different ALTERNATE_INPUT objects within the file. Since each ALTERNATE_INPUT object resides within a different PCF_ENTRY object, the CLASS for an ALTERNATE_INPUT object can always be 1.

CATEGORY is the name of the list of alternates to which the ALTERNATE_INPUT belongs. The PDPS uses CATEGORY to associate different alternates within a list. CATEGORY can be set to any string value of 20 characters or less (e.g., CATEGORY = "Snow Ice"). Alternates that are part of the same list should have matching CATEGORY values.

ORDER is the numerical place that the particular alternate holds in the list of alternates. The first choice or Primary Alternate (with the INPUT_TYPE = "Primary") should have ORDER = 1.

RUNTIME_PARM_ID specifies the Logical ID (in the PCF) for which the PGE will find the Logical ID of the alternate chosen. Since all alternates must be contained within different PCF_ENTRY objects, they all must have different Logical IDs (but all alternates within the same CATEGORY should have the same value of RUNTIME_PARM_ID). The RUNTIME_PARM_ID parameter specifies the Logical ID of a runtime parameter that the PGE may read to find out which alternate was chosen for the particular execution of the PGE.

The TIMER parameter specifies how long to wait for the particular alternate before checking for the next alternate in the list. The parameter value is expressed in the format "<Period Type>=<Length of Period>". Note that “Length of Period” can be specified as a positive integer only. The Alternate Input Production Rule accepts the following “Period Type” values:

- "WEEKS"
  - PDPS should wait for some number of weeks before searching for the next alternate in the list.
  - For example, TIMER = "WEEKS=2" would make PDPS wait two weeks before checking for the next alternate input.
- "DAYS"
  - PDPS should wait for some number of days before searching for the next alternate in the list.
  - For example, TIMER = "DAYS=5" would make PDPS wait five days before checking for the next alternate input.
- "HOURS"
  - PDPS should wait for some number of hours before searching for the next alternate in the list.
  - For example, TIMER ="HOURS=4" would make PDPS wait four hours before checking for the next alternate input.
- "MINS"
  - PDPS should wait for some number of minutes before searching for the next alternate in the list.

  - For example, TIMER = "MINS=5" would make PDPS wait five minutes before checking for the next alternate input.
- "SECS"
  - PDPS should wait for some number of seconds before searching for the next alternate in the list.
  - For example, TIMER = "SECS=2" would make PDPS wait two seconds before checking for the next alternate input.

The WAITFOR parameter specifies whether or not the PGE can be run without the alternate input. Setting WAITFOR = "N" means that the PGE can run without the input if it cannot be found. In a list of alternate inputs, this would have meaning for the last choice only. If WAITFOR = "Y", the PGE is not executed (even after the last alternate timer expires) until one of the alternates in the list can be found.

The TEMPORAL parameter is an unimplemented feature that would allow for searching for alternates from the same time period but a different date. It is currently stored in the PDPS database but is not used.

The OPTIONAL_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = OPTIONAL_INPUT
      .
      .
      .
   END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY
```

The OPTIONAL_INPUT ODL object surrounds an Optional Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Optional Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an OPTIONAL_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- TEMPORAL [not implemented].

The parameters that apply to the Optional Input Production Rule are defined in the same way that the corresponding parameters are defined for the Alternate Input Production Rule. However, note that the Optional Input Production Rule has no WAITFOR parameter. It is irrelevant; in fact, the very essence of the Optional Input Production Rule depends on not “waiting for” the last option but going ahead with the execution of the PGE without the unavailable optional input(s).

Table 11.7.3.2-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs (1 of 2)

```
>OBJECT = PCF_ENTRY
>  CLASS = 16
>  LOGICAL_ID = 1500
>  PCF_FILE_TYPE = 1
>  DATA_TYPE_NAME = "MOD10L2G"      [MODIS Level 2G Snow Cover]
>  DATA_TYPE_VERSION = "1"          [ESDT versioning in release B.0]
>  DATA_TYPE_REQUIREMENT = 1
>  SCIENCE_GROUP = ""
>  OBJECT = FILETYPE
>    CLASS = 1
>    FILETYPE_NAME = "Single File Granule"
>  END_OBJECT = FILETYPE
>  INPUT_TYPE = "Primary"
>  NUMBER_NEEDED = 1
>  OBJECT = ALTERNATE_INPUT
>    CLASS = 1
>    CATEGORY = "Snow Ice"          [User defined]
>    ORDER = 1                      [This data type is sought first]
>    RUNTIME_PARM_ID = 1509         [Run-time parameter holds LID of alternate]
>    TIMER = "HOURS=6"
>    WAITFOR = "N"                  [Force time-out on wait]
>    TEMPORAL = "N"                 [Use most currently produced]
>  END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>  CLASS = 17
>  LOGICAL_ID = 1501
>  PCF_FILE_TYPE = 1
>  DATA_TYPE_NAME = "MOD10A1"       [MODIS Level 3 Daily Gridded Snow Cover data set]
>  DATA_TYPE_VERSION = "1"
>  DATA_TYPE_REQUIREMENT = 1
>  SCIENCE_GROUP = ""
>  OBJECT = FILETYPE
```

Table 11.7.3.2-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs (2 of 2)

```
>     CLASS = 2
>     FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Alternate"
>   OBJECT = ALTERNATE_INPUT
>     CLASS = 2
>     CATEGORY = "Snow Ice"                [User defined]
>     ORDER = 2                           [This data type is sought last]
>     RUNTIME_PARM_ID = 1509              [Run-time parameter holds LID of alternate]
>     TIMER = "HOURS=6"                    [Wait 6 additional hours]
>     WAITFOR = "N"
>     TEMPORAL = "N"
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>   CLASS = 18
>   LOGICAL_ID = 1502
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MIANTASC"       [MISR Terrestrial Atmosphere and Surface
Climatology]
>   DATA_TYPE_VERSION = "1"
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>     CLASS = 3
>     FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Alternate"
>   OBJECT = ALTERNATE_INPUT
>     CLASS = 3
>     CATEGORY = "Snow Ice"                [User defined]
>     ORDER = 3                           [This data type is sought last]
>     RUNTIME_PARM_ID = 1509              [Run-time parameter holds LID of alternate]
>     TIMER = ""                           [Don't wait for this one]
>     WAITFOR = "Y"                        [Start anyway]
>     TEMPORAL = "N"
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
```

11.7.4 Minimum/Maximum Number of Granules Production Rule

The Minimum/Maximum Number of Granules Production Rule makes it possible to specify a range of possible granules for a given input or output for a PGE.

- Inputs.
  - Minimum number of granules the PGE needs for full data coverage.
  - Maximum number of granules for the time period.
- Outputs.
  - Minimum number of outputs that the PGE is expected to produce.
  - Maximum number of outputs that the PGE is expected to produce.

For example, a PGE processes data for every 90-minute interval, has a period of 90 minutes, and takes as input a granule with a period of two hours.

- In many instances one granule of the input will satisfy the PGE.
- In other instances, because of the way the two-hour and 90-minute periods overlap, the PGE needs two input granules to cover the time period.
- Therefore,…
  - Minimum Number of Granules = 1.
  - Maximum Number of Granules = 2.

The Minimum/Maximum Number of Granules Production Rule is different from most production rules because it works for both input and output granules. It allows the PGE to request of a range of inputs (i.e., 1-10 granules), so that it runs with as few as one granule but with as many as ten granules. If a PGE needs at least three granules of a particular input, the minimum number of granules is defined as three and the PGE is not executed until at least three granules are available.

Optional outputs are defined when the Minimum Number of Granules is set to zero. In such cases the PGE can produce none of the particular type of output and still be considered to have executed successfully. If a PGE has a non-zero value for a Minimum Number of Granules associated with an output, and fails to produce any granules of that output type, it is marked as failed.

Figure 11.7.4-1 provides an illustration of the Minimum/Maximum Number of Granules Production Rule. In the example the PGE processes data related to a one-hour period and takes in both Input 1 and Input 2. Since Input 1 has a PERIOD of 1/2 hour, every PGE run requires two Input 1 granules. Input 2 has a PERIOD of 15 minutes, so there are four Input 2 granules for every PGE run.

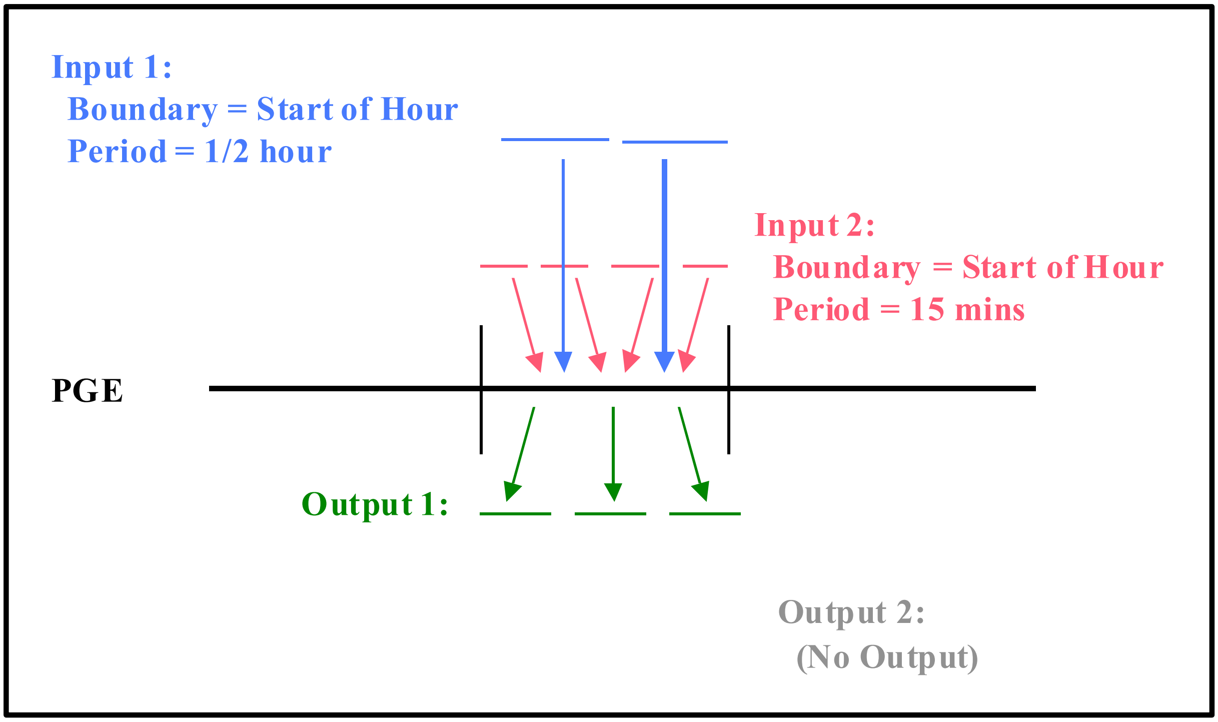


Figure 11.7.4-1. Example of the Minimum/Maximum Number of Granules Production Rule

The PGE produces three Output 1 granules for each run. In this case it does not produce any Output 2 granules.

Minimum and maximum values can affect each input and output as follows:

- Input 1:
  - If Minimum Granules is set to anything equal to or less than two for Input 1, the PGE is scheduled and executed.
  - If Minimum Granules is set to three, the PGE is not scheduled because there are not enough Input 1 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than two for Input 1, the PGE is scheduled and executed.
  - If Maximum Granules is set to one, the PGE is not scheduled because there are too many Input 1 granules (the number exceeds the maximum that the PGE can process).

- Input 2:
  - If the Minimum Granules is set to anything equal to or less than four for Input 2, the PGE is scheduled and executed.
  - If Minimum Granules is set to five, the PGE is not scheduled because there are not enough Input 2 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than four for Input 2, the PGE is scheduled and executed.
  - If Maximum Granules is set to three, the PGE is not scheduled because there are too many Input 2 granules (the number exceeds the maximum that the PGE can process).
- Output 1:
  - If Minimum Granules is set to anything equal to or less than three for Output 1, the PGE is scheduled and executes successfully.
  - If Minimum Granules is set to four, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than three for Output 1, the PGE is scheduled and executes successfully.
  - If Maximum Granules is set to two, the PGE is marked as failed because it produced too many output granules.
- Output 2:
  - If Minimum Granules is set to anything other than zero, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than zero for Output 2, the PGE is scheduled and executes successfully.

The Minimum/Maximum Granules Production Rules are additions to settings/syntax put into the ODL files for other production rules. All Production Rules have a Minimum and Maximum Granule setting for both inputs and outputs, even though both values may be set to one (1).

11.7.4.1 PGE Science Metadata ODL File Parameters

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **input** data includes the following types of entries:

```
OBJECT = PCF_ENTRY
   .
   PCF_FILE_TYPE =
   .
   .
   MIN_GRANULES_REQUIRED =
   MAX_GRANULES_REQUIRED =
   .
   .
   .
END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Minimum/Maximum Production Rule:

- PCF_FILE_TYPE.
- MIN_GRANULES_REQUIRED.
- MAX_GRANULES_REQUIRED.

The PCF_FILE_TYPE parameter is defined by integers in the range of 1 to 8 (inclusive). The integers are codes for the following types of files:

- 1 - product input files.
- 2 - product output files.
- 3 - support input files.
- 4 - support output files.
- 5 - user defined runtime parameters.
- 6 - interim/intermediate input files.
- 7 - interim/intermediate output files.
- 8 - temporary input/output.

For inputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 1, 3 or 6) the following pair of values must be set for each PCF_ENTRY:

- MIN_GRANULES_REQUIRED
  - Minimum number of granules required for the input.
  - A value of zero (MIN_GRANULES_REQUIRED = 0) would mean that the PGE could execute if no granules for that particular input could be found (in effect, the input is an **optional input**).
  - A value of three (for example) would mean that the PGE must have at least three granules of the input before the PGE can be executed.
- MAX_GRANULES_REQUIRED
  - Maximum number of granules for the input that the PGE is able to successfully process.

  - A value of four (for example) would mean that the PGE would process at most four granules for the input.
  - If MAX_GRANULES_REQUIRED = 4 and more than four granules are found for the given input, the PGE is not executed.

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **output** data includes the following types of entries:

```
OBJECT = PCF_ENTRY
   .
   PCF_FILE_TYPE =
   .
   .
   MIN_GRANULE_YIELD =
   MAX_GRANULE_YIELD =
   .
   .
   .
END_OBJECT = PCF_ENTRY
```

For outputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 2, 4 or 7) the following pair of values must be set for each PCF_ENTRY.

- MIN_GRANULE_YIELD
  - Minimum number of granules that the PGE produces for the output.
  - A value of zero (MIN_GRANULE_YIELD = 0) means that the PGE produces no granules for the output (the output is an **optional output**).
  - A value of three (for example) means that the PGE produces at least three granules of the output during a successful execution.
- MAX_GRANULE_YIELD
  - Maximum number of granules that the PGE produces for this output.
  - A value of four (for example) means that at most the PGE produces four granules for the output.
  - Note that sizing of disk space is based on this number, so making it too small could cause problems on the science processor disks.

11.7.5 Optional DPRs Production Rule

The Optional DPRs Production Rule (also called the Data-Scheduled Production Rule) makes the execution of a PGE subject to the availability of a **key input**. The system generates DPRs for every possible instance of the key input data but executes only the DPRs for which data are either produced in data processing or can be acquired from the archive.

The Optional DPRs Production Rule applies to PGEs that process certain kinds of **non-routine data**.

- **Routine Data**
  - Data that can be predicted, that come in at specific intervals and are always of a specified length.
  - Routine data makes it possible for the Basic Temporal Production Rule to schedule PGEs based on their input data.
- **Non-Routine Data**
  - Data that cannot be predicted because they come in at random periods and/or their length is variable.
  - Examples include an "optional" output of an upstream PGE, or data that are archived at random periods (e.g., some forms of ASTER data).

An Optional DPR has as its **key input** a non-routine data type. There are two sets of circumstances that lead to the scheduling of Optional DPRs:

- Every possible time that the input is produced in data processing (i.e., the key input is produced as an "optional" output by an upstream PGE).
- Whenever a new granule (of a particular data type) can be acquired from the archive (e.g., archived data that were inserted at unpredictable times).

An example of the first condition starts with a MODIS PGE that produces a certain product only when the input data were collected during the satellite's "Day" mode. A second MODIS PGE is scheduled to use the optional ("Day"-mode) product from the first MODIS PGE as its key input. The second MODIS PGE is scheduled to run after every instance of the first MODIS PGE; however, only the DPRs that can use the optional products resulting from runs of the first MODIS PGE are executed. The remaining DPRs cannot be executed because there is no input data for them.

The second condition is illustrated by ASTER routine processing, which makes use of the Optional DPRs Production Rule to schedule and execute ASTER PGEs for new data that have been archived. (Note that the DAAC ingests and archives ASTER production data from tapes supplied by the ASTER Ground Data System on a frequent but not entirely predictable basis.) When the Production Planner creates a Production Request for an ASTER PGE, it is necessary to specify the **insertion time** range (i.e., the time period when the desired data were archived) as opposed to the **collection time** (when the satellite instrument gathered the data). DPRs specifying the ASTER PGE are scheduled and executed for the data granules that were actually inserted in the archive during the time period specified in the Production Request.

An illustration of the Optional DPRs production rule is presented in Figure 11.7.5-1. In the figure there are two DPRs (i.e., DPR-1 and DPR-2) for the upstream PGE and two DPRs (i.e., OPT-1 and OPT-2) for the PGE subject to the Optional DPRs Production Rule. The "Optional DPRs" PGE takes as input the optional output of the upstream PGE. When it is executed, DPR-1 produces the optional output, so the dependent DPR (OPT-1) is executed. However, OPT-2 is not executed because DPR-2 (on which OPT-2 depends) does not produce the optional output.

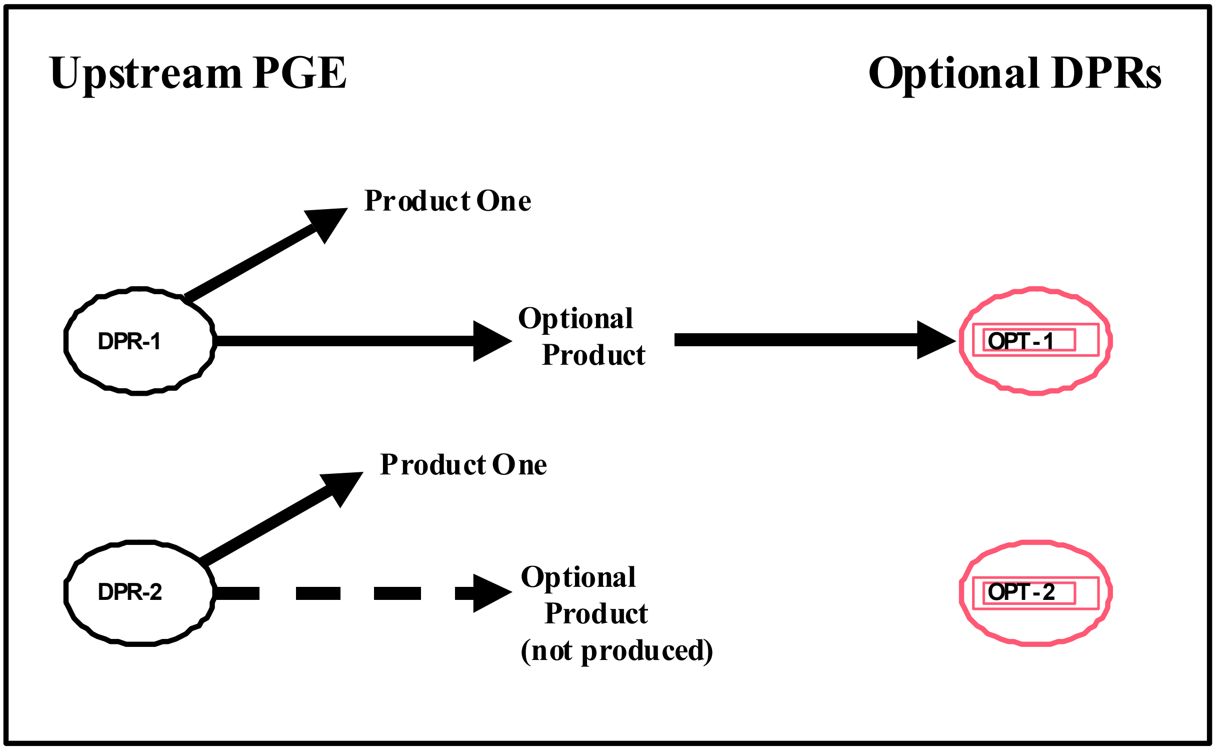


Figure 11.7.5-1. Example of the Optional DPRs Production Rule

The Optional DPRs Production Rule is set up during the SSI&T process. It uses many of the same parameter settings as the Basic Temporal Production Rule so the values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Optional DPRs Production Rule syntax is added.

11.7.5.1 PGE Science Metadata ODL File Parameters

The following two types of PGE science metadata ODL file entries must be made in order to set up the Optional DPRs Production Rule:

- SCHEDULE_TYPE.
- KEY_INPUT.

The SCHEDULE_TYPE parameter is set as follows:

- SCHEDULE_TYPE = “Data”
  - This demonstrates the appropriateness of the term “Data-Scheduled Production Rule.”

  - Other schedule types include Time, Tile, Orbit, and Snapshot.

The key input is designated by including the following parameter in the PCF_ENTRY for whichever input is to be the key input:

- KEY_INPUT = “Y”
  - Assigning a value of “Y” to the KEY_INPUT parameter identifies the data as a key input and it is subsequently treated as such.
  - Either assigning a value of “N” to the KEY_INPUT parameter or leaving out the parameter entirely identifies the non-key input data.
  - Only one key input is allowed per PGE profile.

The Production Planner’s role in the implementation of the Optional DPRs Production Rule was described in the MODIS and ASTER examples previously described and varies with the kind of key input:

- Optional output of an upstream PGE (MODIS example).
  - Production Planner creates Production Requests for the PGE subject to the Optional DPRs Production Rule and specifies the same date/time range as for the upstream PGE.
  - Some of the DPRs generated as a result of the Production Request will never run due to lack of input data.
- Ingested on an irregular time schedule (ASTER example).
  - Production Planner specifies the data **insertion time** range when creating Production Requests.
  - All DPRs generated as a result of the Production Requests should be capable of running.

11.7.6 Intermittent Activation Production Rule

The conditions for executing most PGEs are well defined. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g., run once for every granule or run monthly averages once a month). However, some PGEs have additional or different constraints on when they are run.

A PGE can be set up to run on every n^{th} instance of input data. For example, a QA PGE that is run on a daily product may need to be run only every fifth day to provide a spot check. Note that this does **not** refer to the common case of running a weekly averaging PGE only once each week, which would be handled by the Basic Temporal Production Rule and the time ranges specified for the input and output ESDTs. Rather, this is a special case where a PGE **can** be run every day (or hour, week, etc.), but for some reason (such as a QA check) it is desired to run the PGE only every n^{th} day.

To implement the Intermittent Activation Production Rule the Production Planner supplies the following information (via the Production Request Editor) when creating a production request:

- **Number to Skip**
  - Number of DPRs to be skipped (not executed).
  - Entered in the **Skip** field on the Production Request Editor.
- **Number to Keep**
  - After skipping the specified number of DPRs, how many are to be kept?
  - Entered in the **Keep** field on the Production Request Editor.
    - The number to keep is usually one but could be any number.
- **Skip First**
  - Button on the Production Request Editor.
  - Selected to skip the first DPR.
    - Not selected if the first DPR is to be run.

The Planning Subsystem uses the preceding information to establish a pattern of execution. The pattern is effective for the single PR in which the “number to skip” and the “number to keep” are specified; it is not maintained between PRs.

The following example of the Intermittent Activation Production Rule is shown in Figure 11.7.6-1:

- The Production Planner prepares a production request for a 14-day period, generating 14 DPRs.
- The Production Planner made the following selections on the Production Request Editor:
    - Entered “4” in the **Number to Skip** field.
    - Entered “1” in the **Number to Keep** field.
    - Did **not** select the **Skip First** button.
- Consequently, the following results are obtained:
    - First DPR runs. Four DPRs (second through fifth) are skipped.

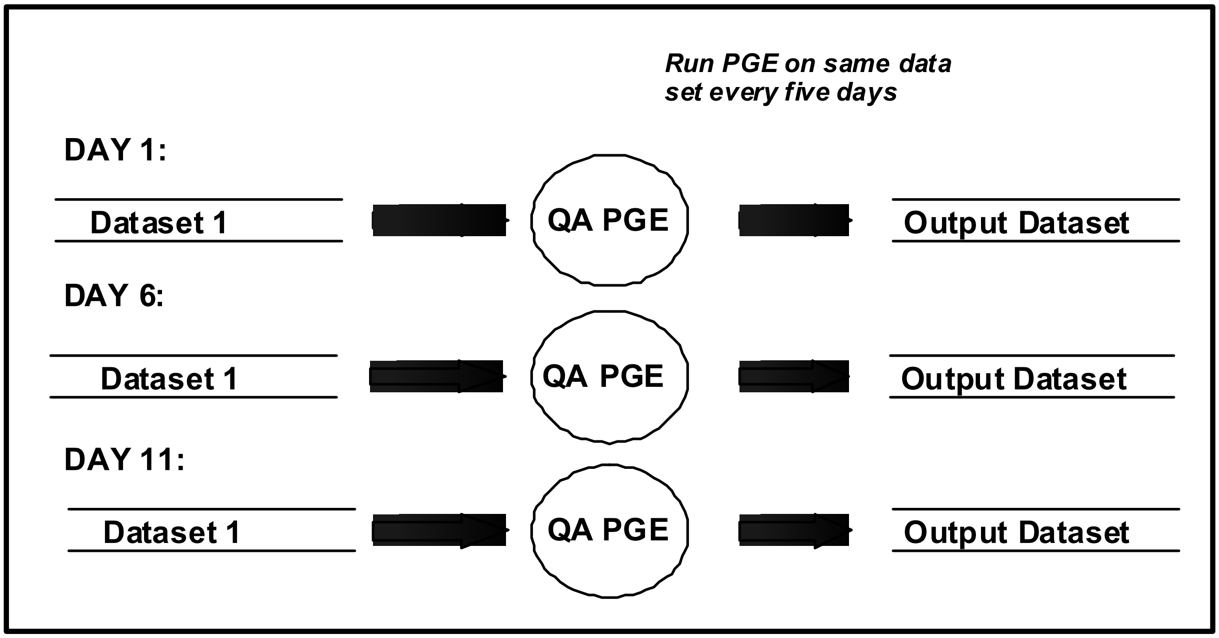


Figure 11.7.6-1. Example of the Intermittent Activation Production Rule

- Sixth DPR runs.
- Four DPRs (seventh through tenth) are skipped.
- Eleventh DPR runs.
- Remaining three DPRs (twelfth through fourteenth) are skipped.

11.7.7 Metadata Checks and Metadata Query Production Rules

The Metadata Checks and Metadata Query Production Rules are similar in definition and use. Both production rules allow the PGE to specify granule-level metadata values that define whether the PGE can accept one (or more) of its inputs. The rules differ only in the results of metadata search performed.

- Metadata Checks Production Rule.
  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server "checks" the metadata of all granules that match the time frame with respect to the value(s) allowed by the PGE.
  - If any granule fails to match the specified value(s), the PGE is not executed.
- Metadata Query Production Rule.

  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server adds to the query the metadata value(s) desired by the PGE.
  - Only the granules that match the time frame of the PGE plus the granule-level metadata value(s) specified by the PGE are staged for the PGE to use as input.
  - If no granules are found matching the conditions and the input is not optional, the PGE is not executed.
- Example of Metadata Checks:
  - A MODIS PGE is run when the Percent Cloud Cover of its inputs is greater than 25 percent.
  - The Metadata Checks Production Rule is used to specify the granule-level metadata value of greater than 25.
  - When the PGE is scheduled and is ready to start, two granules match the timeframe of the Production Request for the input with the Metadata Check.
  - If both granules have a Percent Cloud Cover greater than 25 percent, execution of the PGE starts and both granules are staged.
  - If one of the granules has a Percent Cloud Cover of 15 percent, the PGE is not executed.
- Example of Metadata Query:
  - A MODIS PGE is run when as many granules as possible of one of its inputs have a QA Value = "Good".
  - The Metadata Query Production Rule is used to specify the granule-level metadata value = "Good".
  - When the PGE is scheduled and is ready to start, two granules match the time frame of the production request for the input with the Metadata Query.
  - If both granules have a QA Value = "Good", execution of the PGE starts and both granules are staged.
  - If one of the granules has a QA Value = "Bad", the PGE executes but with only one granule (the one with QA Value = "Good").

The Metadata Checks and Metadata Query Production Rules are used in conjunction with the times specified in the Basic Temporal Production Rule or other production rules. The Metadata Check or Query is added information that further refines what granules are sought by the PGE.

Multi-Granule ESDTs are a special case of the Metadata Query Production Rule. Multi-Granule ESDTs are used for PGE inputs or outputs when more than one granule of the same ESDT exists for the same temporal range (time period). The Multi-Granule ESDT mechanism employs a metadata parameter to differentiate between the "equal in time" granules. A metadata parameter is selected that is unique across granules for the same time period and that is used by PDPS to keep track of which granule is which when the granules are produced. Later, if only one of a pair of granules for a particular time period is needed as input to the PGE, the Metadata Query is used to ensure that PDPS schedules the correct granule as input.

The **Data Day Production Rule** is actually an addition to the Metadata Query Production Rule involving runtime parameter values. There is a pair of settings (Start Data Day and End Data Day) that allow a PGE to perform a Metadata Query for the start of the Data Day and the end of the Data Day. A separate section of this lesson is devoted to the Data Day Production Rule.

Using runtime parameter values is a capability of the Metadata Query and Metadata Checks Production Rules. Rather than use a hard-coded value for the check or query, a value computed from one of the other production rules can be used.

Figure 11.7.7-1 illustrates the Metadata Checks and Metadata Query Production Rules. If no Metadata Check or Query were applicable, the PGE shown in the figure would use three granules of input (i.e., Granules A through C). However, let us assume that the metadata value to be checked/queried is %CloudCover. Each granule has a different value for %CloudCover.

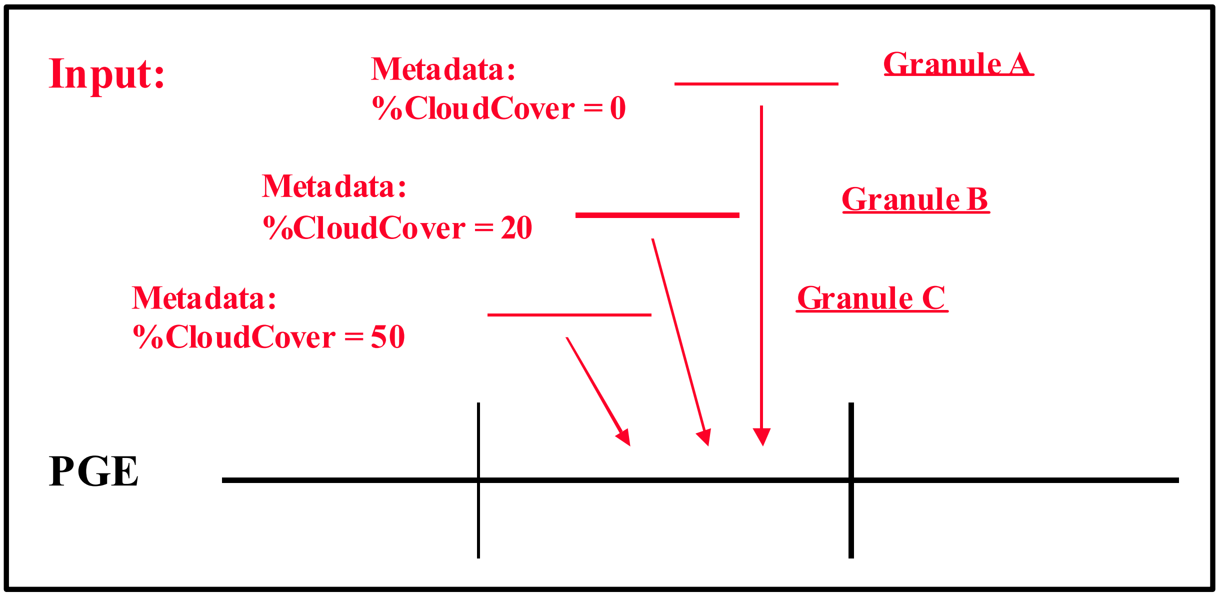


Figure 11.7.7-1. Example of the Metadata Checks and Query Production Rules

The following results demonstrate the differences between the Metadata Checks and Metadata Query Production Rules, especially with respect to the number of inputs that the PGE receives when different values are specified:

- Metadata Check of %CloudCover < 80:
  - In this case all three granules are acquired and the PGE is scheduled and executed.
- Metadata Query of %CloudCover < 80:
  - All three granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 50:
  - The PGE is not scheduled because only one of the three granules (Granule C) meets the criterion.
- Metadata Query of %CloudCover = 50:
  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, that one granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover < 50:
  - The PGE is not scheduled because only two of the three granules (Granule A and B) meet the criterion.
- Metadata Query of %CloudCover < 50:
  - Granules A and B are found and if the PGE's Min/Max Granules parameters are set to allow two granules, the granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover <= 50:
  - The PGE is scheduled and executed because all three granules meet the criterion.
- Metadata Query of %CloudCover <= 50:
  - All three granules are found and acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 20:
  - The PGE is not scheduled because only one of the three granules (Granule B) meets the criterion.
- Metadata Query of %CloudCover = 20:
  - Granule B is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover < 20:
  - The PGE is not scheduled because only one of the three granules (Granule A) meets the criterion.

- Metadata Query of %CloudCover < 20:
  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 10:
  - The PGE is not scheduled because none of the three granules meets the criterion.
- Metadata Query of %CloudCover = 10:
  - The PGE is not scheduled because no granules are returned from the query (unless Minimum Granules is set to 0).

Note that there can be more than one Metadata Check or Metadata Query on a given input. In the preceding example, a Metadata Check on %CloudCover can be combined with a Metadata Query on another parameter to further limit the input.

The Metadata Checks and Metadata Query Production Rules are additions to settings/syntax put into the ODL files for other production rules. The addition of a Metadata Check or a Metadata Query to an input means that other production rules used to evaluate that input will be applied in combination with the Metadata Check or Metadata Query.

11.7.7.1 PGE Science Metadata ODL File Parameters

Although the Metadata Checks and Metadata Query Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY in the PGE science metadata ODL file; i.e., the METADATA_CHECKS object and the METADATA_QUERY object.

The METADATA_CHECKS object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = METADATA_CHECKS
      .
      .
      .
   END_OBJECT = METADATA_CHECKS
END_OBJECT = PCF_ENTRY
```

The METADATA_QUERY object has the same syntax except "METADATA_QUERY" replaces "METADATA_CHECKS" in every instance.

Most of the following parameters must be set in the PGE science metadata ODL file within the METADATA_CHECKS or METADATA_QUERY ODL object (as applicable) in order to implement either the Metadata Checks or Metadata Query Production Rule:

- CLASS.
- PARM_NAME.
- OPERATOR.
- VALUE.
- DATABASE_QUERY.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Checks or Metadata Query objects within the file. Since each Metadata Checks or Metadata Query object resides within a different PCF_ENTRY object, the CLASS for an METADATA_CHECKS or METADATA_QUERY object can always be 1 (e.g., CLASS = 1).

PARM_NAME is the name of the metadata parameter on which the check or query is to be performed. The value specified for PARM_NAME (e.g., PARM_NAME = “%CloudCover”) must be part of the granule-level metadata of the ESDT. In addition, it must match the parameter name specified in the ESDT science metadata ODL file.

OPERATOR is the operator (e.g., OPERATOR = “==”) on which the check/query is to be performed. The following values are valid for OPERATOR:

- ">"
  - Value in metadata must be greater than.
- "<"
  - Value in metadata must be less than.
- ">="
  - Value in metadata must be greater than or equal to.
- "<="
  - Value in metadata must be less than or equal to.
- "=="
  - Value in metadata must be equal to.
- "!="
  - Value in metadata must be **not** equal to.

VALUE is the value (e.g., VALUE = 50) against which the metadata parameter (defined by PARM_NAME) is compared (using the operator specified by the OPERATOR parameter). The value for the VALUE parameter should be the type of data (e.g., integer, string) as defined in the ESDT ODL metadata for the parameter.

DATABASE_QUERY indicates whether the value for the Metadata Check or Query should be retrieved from the PDPS database rather than through the use of the VALUE parameter.

Specifying DATABASE_QUERY permits **runtime parameter values** to be used for Metadata Query or Metadata Checks. The following values are valid for the DATABASE_QUERY parameter:

- "NONE"
  - Use the value in the VALUE parameter; no value from the PDPS database is used.
- "PATH NUMBER"
  - Use the Path Number (0-233) of the orbit for which the PGE is scheduled.
- "ORBIT NUMBER"
  - Use the Orbit Number of the orbit for which the PGE is scheduled.
- "TILE ID"
  - Use the Tile ID of the current Data Processing Request.
- "START DATA DAY"
  - Use the Start Data Day for the current Data Processing Request.
- "END DATA DAY"
  - Use the End Data Day for the current Data Processing Request.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_NAME (e.g., KEY_PARAMETER_NAME = "ParameterName" for metadata checks or qeuries within the MeasuredParameters group) in conjunction with the KEY_PARAMETER_VALUE allows PDPS to determine which container within the multi-container group is to be the object of the check or query. KEY_PARAMETER_NAME is **not** used for product-specific attributes.

KEY_PARAMETER_VALUE is an optional parameter that is used to specify the **value** (e.g., KEY_PARAMETER_VALUE = "LandCoverage") for the container within a multi-container metadata group (i.e. the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the PGE science metadata ODL file and ESDT science metadata ODL file must match.

Multi-Granule ESDTs are created by adding the following parameter to the PCF_ENTRY in the PGE science metadata ODL file:

- DISTINCT_VALUE.

The DISTINCT_VALUE must be set to the value of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. In addtion, the input or output defined by the PCF entry must have a corresponding DISTINCT_PARAMETER entry in the ESDT science metadata ODL file.

11.7.7.2 ESDT Science Metadata ODL File Parameters

The METADATA_DEFINITION ODL object surrounds the definition for Metadata Checks or Metadata Query information within the ESDT science metadata ODL file. An OBJECT/END_OBJECT pair is needed to separate the parameters defining the Metadata Definition from the rest of the parameters defining the ESDT with the following syntax:

```
OBJECT = METADATA_DEFINITION
   .
   .
   .
END_OBJECT = METADATA_DEFINITION
```

A METADATA_DEFINITION object can match multiple Metadata Checks or Metadata Query objects in various PGE science metadata ODL files. There is no difference between the two production rules with respect to the parameters that need to be set in the ESDT science metadata ODL file. Most of the following parameters must be set:

- CLASS.
- PARM_NAME.
- CONTAINER_NAME.
- TYPE.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Definition objects within the file. Each Metadata Definition object within the file must have a **different** CLASS value.

PARM_NAME is the name of the Metadata parameter on which the check or query will be performed. The value specified for PARM_NAME must be part of the granule-level metadata of the ESDT. It must also match the parameter name specified in the PGE science metadata ODL file(s).

CONTAINER_NAME is the name of the Metadata Group within which the metadata parameter defined by PARM_NAME is contained. For product-specific attributes CONTAINER_NAME is set to the string "AdditionalAttibutes" (i.e., CONTAINER_NAME = "AdditionalAttibutes").

TYPE indicates the type of data within the metadata parameter. The following values are valid for TYPE:

- "INT"
  - Integer data.

- "FLOAT"
  - Floating point data.
- "STR"
  - String or character data.
  - Note that dates and times are considered string data.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_NAME allows PDPS to determine which container within the multi-container group is to be the object of the check or query.

KEY_PARAMETER_VALUE is an optional parameter that is used to specify the value for the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the ESDT science metadata ODL file and PGE science metadata ODL file must match.

The ESDT science metadata ODL file for an input specifying Multi-Granule ESDTs needs to have the following parameter added:

- DISTINCT_PARAMETER.

The DISTINCT_PARAMETER must be set to the name of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. A corresponding METADATA_DEFINITION must be created to help PDPS find the specified metadata parameter when querying the Science Data Server.

11.7.8 Data Day Production Rule

The Data Day Production Rule is an addition to the Metadata Query Production Rule involving runtime parameter values. The Data Day Production Rule uses a query to the PDPS database for the time period for the DPR and a Metadata Query for data matching the Data Day. The Data Day is defined as a day within twelve hours of the current day. There is a pair of settings (Start Data Day and End Data Day) that provide parameters for the Metadata Query.

The Start Data Day and End Data Day values are calculated by subtracting twelve hours from the starting day for which the PGE is executing and adding twelve hours onto the ending day for which the PGE is running.

- Data Day for a PGE is running on data from 07/04 00:00:00 to 07/05 00:00:00 is defined as follows:
  - START_DATA_DAY = 07/03 12:00:00
  - END_DATA_DAY = 07/06 12:00:00.

11.7.9 Spatial Query Production Rule

The Spatial Query Production Rule allows a PGE to select input(s) based on the spatial coverage of another input (called the **key input**). The PDPS queries the Science Data Server for the spatial coverage of the key input, then uses it in acquiring any subsequent inputs that the PGE has requested that have the same spatial coverage.

- Example:
  - Level 0 input data for an ASTER DPR covers a small section of the Earth.
  - The PGE requires ancillary data that covers the same area to complete its processing.
  - The PGE uses the Spatial Query Production Rule to mark the geographic input as its key input.
  - The PGE specifies that the ancillary input is to be retrieved for the same spatial coverage as that of the key input.
  - When PDPS finds an input granule for the PGE, it performs a Spatial Query to acquire the ancillary input with the same spatial coverage as that of the key input.

Without specifying coordinates, PDPS can match inputs against the spatial constraint of the key input, and give to a PGE only those granules which overlap in area.

For Release 5B Spatial Pad will be added to the Spatial Query Production Rule. Spatial Pad is a means of padding the spatial constraints of the key input. The specified pad is added to all sides of the key input's spatial shape. All granules that intersect the expanded area are retrieved.

Figure 11.7.9-1 is an illustration of the Spatial Query Production Rule. The figure shows a PGE that has two input types, one of which is the key input. The other type of input has granules labeled with the names of various colors. One granule (i.e., “green”) of the key input is found. The spatial coordinates of the granule are retrieved and all inputs of the second ESDT are checked for overlap with the key input’s coordinates.

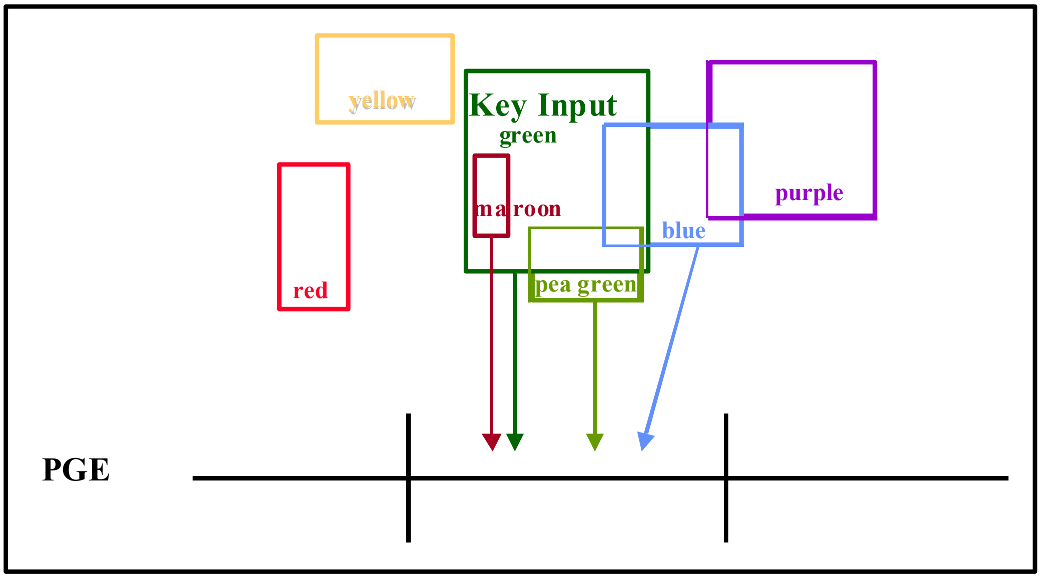


Figure 11.7.9-1. Example of the Spatial Query Production Rule

Assuming that all granules relate to the same time period, the granules are evaluated as follows:

- The “yellow” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The “red” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The “blue” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint is within the constraint of the key input.
- The “maroon” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. The spatial constraint of this granule is completely within the constraint of the key input.
- The “pea green” granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint overlaps with that of the key input.
- The “purple” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input. It does not matter that it overlaps with another input that is accepted (i.e., the “blue” granule).

The Spatial Query Production rule is somewhat of an addition to other production rules. As such, it needs the same parameter settings as the Basic Temporal Production Rule. The values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Spatial Query Production Rule syntax is added.

11.7.9.1 PGE Science Metadata ODL File Parameters

In order to implement the Spatial Query Production Rule the following two parameters must be defined in the applicable PCF_ENTRY (each input is defined by a separate PCF_ENTRY in the PGE science metadata ODL file):

- KEY _INPUT.
- QUERY_TYPE.

The entries are made in the following format:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   QUERY_TYPE = “Spatial”
   KEY_INPUT = “Y”
   .
   .
   .
```

```
            END_OBJECT = PCF_ENTRY
```

QUERY_TYPE indicates what type of query is to be done to acquire the input defined by the PCF_ENTRY object. Valid values are as follows:

- "Temporal" - Input is acquired based on time.
  - The Basic Temporal and/or the Advanced Temporal Production Rules is/are used to get the input.
  - “Temporal” is the value that is assumed if the parameter is left out of the PCF_ENTRY object.
- "Spatial" - Input is acquired based on spatial coordinates (as well as time).
  - An input must be designated the key input to be used in determining the spatial constraints of the search.
  - “Spatial” is the value specified for each input that uses the Spatial Query Production Rule.
- "Tile" - Input is acquired by the spatial definition of a tile.
  - Refer to the Tiling Production Rule for additional information.
- "Already Created Tile" - Input is acquired based on the tile ID of an already created tile.
  - Refer to the Tiling Production Rule for additional information.

The KEY_INPUT is the input on which the spatial queries for other inputs will be based. When a KEY_INPUT parameter is assigned a value of “Y” the corresponding input is designated a key input and is treated as such. A value of “N” or leaving out the parameter entirely specifies a non-key input. Only one (1) key input is allowed per PGE Profile.

11.7.10 Tiling Production Rule

The Tiling Production Rule allows a PGE to run over a series of specific geographic locations called "tiles". The tiles are defined before the PGE is scheduled, specifying the longitude and latitude of four points that outline each tile. When the PGE is scheduled, it is scheduled for an entire day, and data is queried based on both a timeframe and the geographic location specified. Each run of the PGE for that day is for a specific tile, and only data that overlap or fit within the geographical coordinates of the tile are staged for the PGE.

- Example:
  - A MODIS PGE is designed to run on data for a specific geographic location every day.
  - The location is expressed as a polygon defined by latitude and longitude coordinates.
  - The MODIS PGE is scheduled every day, and data are retrieved that match the time period (the day for which the PGE is being executed) and some part of it falls within the geographic constraints of the tile.

  - The PGE runs and produces data that define information about the particular tile.

Period and **boundary** are used to specify the timing of input data and provide indications of how often the PGE should be executed. But at least some of the input data are retrieved on the basis of the coordinates defined for the tile on which the PGE is executing. In fact there are really two kinds of tiling:

- The PGE takes in data based on geographic shapes (tiles) and produces an output or outputs for the specified geographical coverage.
- The PGE takes in an already tiled product as input.
  - This form of tiling is more like a Metadata Query using a runtime parameter value to acquire the correct tiled data.

There are some possible future enhancements to the Tiling Production Rule but they have not been scheduled yet:

- **Zonal Tiling** supports tiles that cover a band around the Earth between two given latitudes.
- **Tile Clustering** involves grouping tiles that cover nearby geographic locations together so that data that span the tiles may be staged only once.
  - Intended to improve the performance of Tiling.
  - Also provides for the ability to prioritize one group of tiles over others (so specific geographic outputs are produced before other geographic outputs).

Runtime parameters can be set to the ID of the tile being processed. Since PDPS schedules a Tiling PGE to run once per tile, it can pass the identifier of the tile to the PGE. The identifier can be placed under a specified runtime parameter in the PCF, or it can be used in a Metadata Query for a PGE that would use already tiled data as input.

Figure 11.7.10-1 provides an example of the Tiling Production Rule. The PGE runs once per defined tile. So for every tile in the Tile Scheme a Data Processing Request is created to run using data that match the geographic extent of the tile. The PDPS sends the coordinates of the tiles (e.g., Tiles 1 through 3 in Figure 11.7.10-1) to the Science Data Server when requesting data and acquires only the granules that fall fully or partially within the defined tile.

The PGE itself must be set up to handle the fact that the entire area of the tile may not be covered by available data. In addition, because PDPS does not keep track of tiles once they have been produced, the PGE must set the metadata of the output products so a downstream Tiling PGE can acquire the correct granules for a given tile. The PDPS matches up the granules needed for a downstream PGE via a query to the Data Server Subsystem.

11.7.10.1 Tiling Based on Already Tiled Data

As previously stated, the second form of Tiling concerns PGEs based on tiles that have already been created by other PGEs. Tiling based on already tiled data is really a combination of the Metadata Query Production Rule and the Tiling Production Rule. The latter is used in running the PGE(s) once per tile, just like any other Tiling PGE. The Metadata Query Production Rule is used in acquiring the previously tiled data by querying the Science Data Server for metadata that match the tile ID that is currently being executed. The query depends on the "runtime parameters" function of Tiling to provide the tile ID relevant to the PGE that is currently being executed.

The Tiling Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. A PGE that performs Tiling still needs a **boundary** and **period** and other such parameters. The difference is that values specified for some of the fields provide Tiling information. Furthermore, Tiling requires that a tile scheme be identified in the PGE science metadata ODL file. The tile scheme is defined in a tile science metadata ODL file.

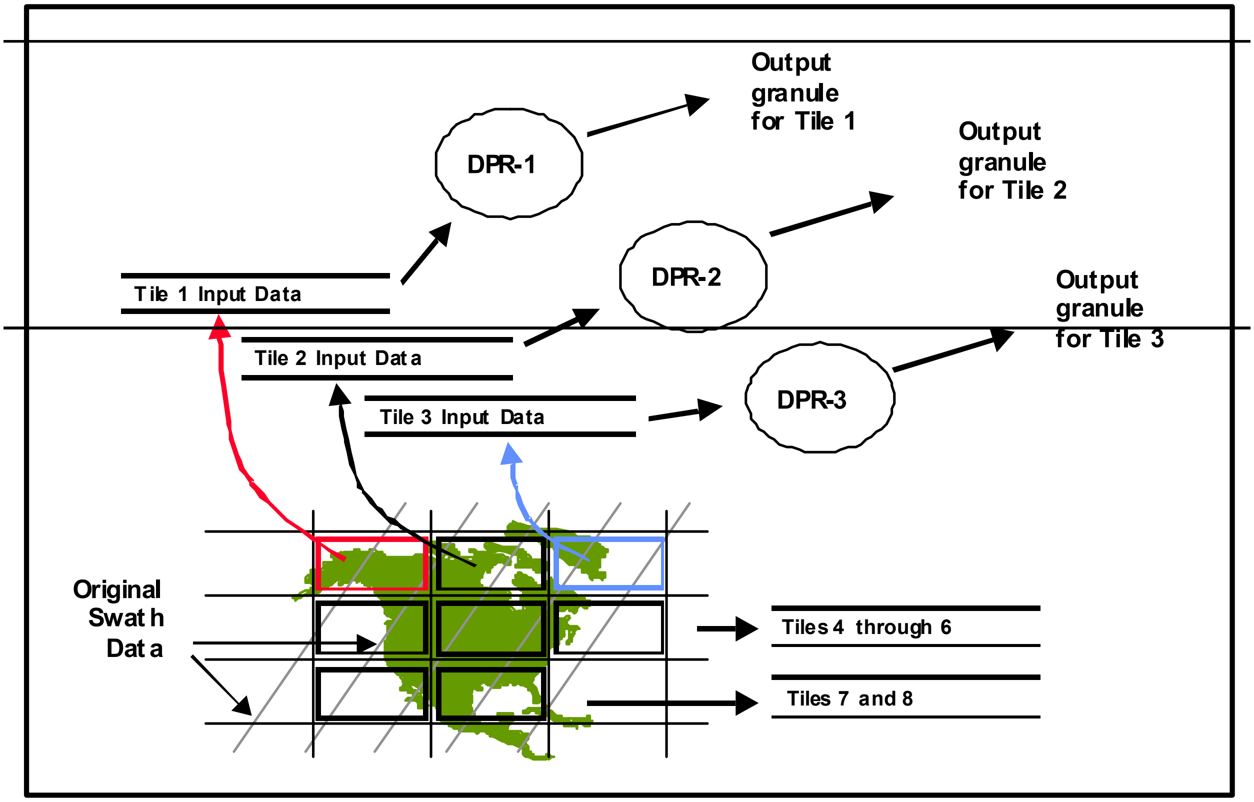


Figure 11.7.10-1. Example of the Tiling Production Rule

11.7.10.2 PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Tiling Production Rule:

- SCHEDULE_TYPE.
- TILE_SCHEME_NAME.

In addition, the following parameter is used within a PCF_ENTRY when defining the Tiling Production Rule:

- QUERY_TYPE.

The SCHEDULE_TYPE parameter defines the type of scheduling that will be done for the PGE. Values for the Tiling Production Rule are:

- "Tiling"
  - Tile-Scheduled.
  - The PGE is scheduled based on the specified PROCESSING_PERIOD and PROCESSING_BOUNDARY, but a DPR is created for each defined tile.

The TILE_SCHEME_NAME parameter is the name of the Tile Scheme to be used by PDPS when scheduling and executing PGEs for each defined tile. There must be a tile ODL file that matches the specified scheme name.

The QUERY_TYPE parameter specifies the type of query to be performed on the input defined by the PCF_ENTRY Object. It uses the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   QUERY_TYPE =
   .
END_OBJECT = PCF_ENTRY
```

For Tiling PGEs there are two possible values for QUERY_TYPE:

- "Tile"
  - The data for the input are acquired on the basis of the spatial constraints of the current tile.
  - Used for a PGE that takes in raw data and produces one or more tiles of data.
- "Already Created Tile"
  - The input is a tiled output of another Tiling PGE.
  - Used for a PGE that takes input from one or more other Tiling PGEs.
  - A Metadata Query must be added to this PCF_ENTRY in order for the correct tiled input to be acquired.

11.7.10.3 Tile Science Metadata ODL File Parameters

The following parameter must be set in the Tile science metadata ODL file in order to implement the Tiling Production Rule:

- TILE_SCHEME_NAME.

In addition, the following ODL objects are used within a PCF_ENTRY to define the Tiling Production Rule:

- TILE object.
- TILE_COORDINATE object.

The TILE_SCHEME_NAME parameter identifies the tile scheme for which the tile information is being specified. Values are limited by the following constraints:

- The string specified can be no more than 20 characters.
- The string specified should match the string specified for TILE_SCHEME in the PGE science metadata ODL file.

The TILE object is an ODL object that surrounds each tile definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each tile that is going to be expressly defined:

```
OBJECT = TILE
    .
    .
    .
END_OBJECT = TILE
```

The following parameters are set in the TILE object in order to implement the Tiling Production Rule:

- CLASS.
- TILE_ID.
- TILE_DESCRIPTION.

CLASS is a simple counter used to differentiate the different TILE objects within the file. Each TILE object needs to have a different CLASS value.

TILE_ID is the tile identifier for the tile being defined. The TILE_ID must be an integer (e.g., TILE_ID = 12) and must be greater than zero but less than the maximum integer. If a Tile ID is defined in other tile schemes, it must have the same coordinates and description.

TILE_DESCRIPTION is a string of characters (255 characters maximum) that describes what the tile is for, such as its geographic location or area that it covers (e.g., TILE_DESCRIPTION = "Upper North America").

The TILE_COORDINATE object is an ODL object that defines a coordinate (latitude and longitude) for a tile. An OBJECT/END_OBJECT pair is needed for each coordinate that is defined. Each tile must have four TILE_COORDINATE objects defined. (Currently only four-sized polygons are allowed; however, a possible future enhancement would provide for polygons with more than four points.) Coordinate objects must follow a clockwise sequence so that if lines were drawn between the points in the order they are given the desired shape would be drawn.

Coordinate objects conform to the following format:

```
OBJECT = TILE
   .
   .         .
   OBJECT = TILE_COORDINATE
      CLASS =
      LATITUDE =
      LONGITUDE =
   END_OBJECT = TILE_COORDINATE
   .
   .         .
END_OBJECT = TILE
```

The following parameters are set in the TILE_COORDINATE object in order to implement the Tiling Production Rule:

- CLASS.
- LATITUDE.
- LONGITUDE.

The CLASS parameter (e.g., CLASS = 1) is an object counter that is used only to distinguish objects. The value assigned to CLASS must be an integer greater than zero and must be unique in the file for the particular type of object.

The LATITUDE parameter (e.g., LATITUDE = 12.15) describes the latitude component of the tile coordinate. There is one LATITUDE entry per TILE_COORDINATE object.

The LONGITUDE parameter (e.g., LONGITUDE = -43.22) describes the longitude component of the tile coordinate. There is one LONGITUDE entry per TILE_COORDINATE object.

11.7.11 Closest Granule Production Rule

The Closest Granule Production Rule allows a PGE to request the nearest input granule from the Data Processing Request time. The PDPS requests a search forward or backward for a specified period of time until it finds a granule that matches the request. However, there is a limit to the number of queries that are performed. The number of queries and the period length of the query are specified during SSI&T.

- Example:
  - A PGE processes data at daily intervals and could use a particular type of calibration granule that would allow it to determine the nearest parameters of the instrument.
  - Although most calibration coefficients are defined as static granules, in this case there is a dynamic granule that is received about once a month.
  - The closest such granule would be optimum, so the PGE uses the Closest Granule Production Rule to search forward or backward from the time of the DPR to find the nearest calibration granule.

The Closest Granule Production Rule supersedes the Most Recent Granule Production Rule. The latter allowed the search for inputs to go backward in time from the start of the DPR. The Closest Granule Production Rule allows the search for input granules to go either backward or forward in time, increasing the flexibility of the rule. The Closest Granule Production Rule has all of the ability of Most Recent Granule, plus the ability to search forward in time for input data.

The Closest Granule Production Rule uses two values to determine the period of the query. The two values are concerned with the direction of the query and the number of queries allowed.

- Offset.
  - Tells the PDPS software the query duration.
  - The sign (+ or -) indicates whether the query goes forward (positive) or backward (negative) in time.
- Retries.
  - Tells the PDPS software how many time periods (as defined by the offset) to search (either forward or backward) in time for matching granule.

The PDPS does a Basic Temporal query before using Closest Granule to find the input. If the desired input is not found within the time period of the DPR, PDPS performs a query against the Science Data Server for the period defined by the offset. Again, if no matching granule is found, PDPS repeats the query, going backward or forward in time by the value specified in the offset. If no acceptable granule has been found before the maximum number of queries is reached, PDPS fails to generate the DPR due to insufficient input data.

Figure 11.7.11-1 illustrates the Closest Granule Production Rule. In the example, the PGE has a boundary of “start of day” and a period of one hour, so it is scheduled to run for one hour’s worth of input data. The input has a period of one hour, and can come in at any hour of the day. Consequently, the PGE requests one granule of input.

The PGE has defined the Closest Granule Production rule with a –6-hour period of the query, meaning that it queries back in time in six-hour intervals. The number of retries is two. The PDPS performs a query for the input based on the time period of the DPR. Not finding any matching data, it uses the Closest Granule information to query for a six-hour period beginning six hours before the start time of the DPR. Again nothing is found, so a second Closest Granule query is performed, this one six hours before the last Closest Granule query. The second query results in the discovery of two matching granules. The PDPS selects the granule that is later in time and schedules the PGE to use it as input.

If the Closest Granule Production Rule were used in conjunction with the Minimum/Maximum Number of Granules Production Rule, it might be possible for both granules to be selected in the previously described Closest Granule query. If the example included setting the Maximum Number of Granules to two, both granules would be selected as input to the PGE.

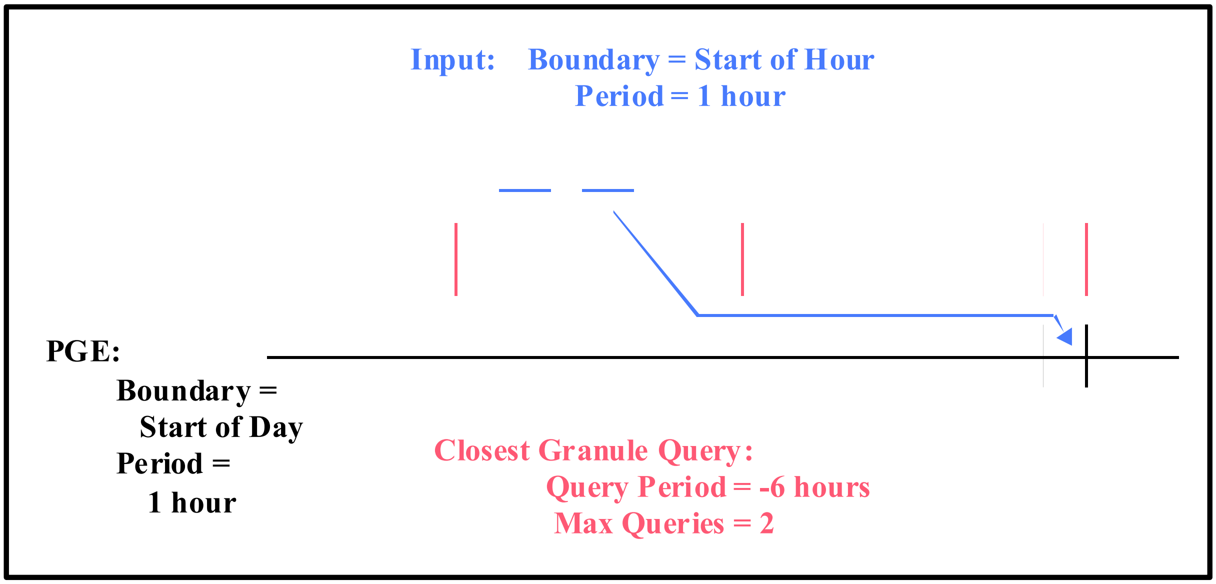


Figure 11.7.11-1. Example of Closest Granule Production Rule

The Closest Granule Production Rule needs the same parameter settings as the Basic Temporal Production Rule. The values needed for the Basic Temporal Production Rule must be set before the Closest Granule Production Rule syntax is added.

11.7.11.1 PGE Science Metadata ODL File Parameters

In addition to the parameter settings for the Basic Temporal Production Rule, the following parameters must be set within the appropriate PCF_ENTRY in the PGE science metadata ODL file in order to implement the Closest Granule Production Rule:

- CLOSEST_QUERY_OFFSET.
- CLOSEST_QUERY_RETRIES.

CLOSEST_QUERY_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR and uses as the query for the requested input data type. The specified value has the format "<Period Type>)=<Length of Period>" (e.g., CLOSEST_QUERY_OFFSET = “HOURS=6”). The following “Period Type” values are used in implementing the Closest Granule rule:

- "WEEKS"
  - Offset is some number of weeks.
  - For example, "WEEKS=2" would be a 14-day offset.

- "DAYS"
  - Offset is some number of days.
  - For example, "DAYS=5" would be a 120-hour offset.
- "HOURS"
  - Offset is some number of hours.
  - For example, "HOURS=4" would be a 240-minute offset.
- "MINS"
  - Offset is some number of minutes.
  - For example, "MINS=5" would be a 300-second offset.
- "SECS"
  - Offset is some number of seconds.

CLOSEST_QUERY_RETRIES is the maximum number of Closest Granule queries before the DPR fails due to insufficient input data. The specified value is an integer value that is limited only by the maximum size of an integer on the executing hardware.

Note that the longer the offset value or the greater the number of retries, the more time that each query requires due to search time at the Science Data Server and processing time of any granules returned. The combination of a large offset with a large number of retries, can (if no data granules are found) consume a lot of time while failing to generate a DPR.

11.7.12 Orbital Processing Production Rule

The Orbital Processing Production Rule is similar to the Basic Temporal Production Rule in that both define the time period for the inputs and outputs of the PGE. The difference is that the Orbital Processing Production Rule uses the orbit of the spacecraft to determine that time period. A PGE that processes data related to every orbit of a satellite uses data related to a time period that is computed from the orbit of that satellite.

- Example:
  - A PGE processes Level 0 data related to each orbit of the AM-1 satellite.
  - The AM-1 satellite has an orbital period of 98 minutes so the PGE is scheduled to process data for each 98-minute interval.
  - Since Level 0 data are received every two hours, the data staged for the PGE include every Level 0 granule that falls within the 98 minute PGE interval.
  - Only one granule of Level 0 data is relevant to some 98-minute orbits.
  - Two granules of Level 0 data are relevant to other 98-minute orbits.

The Orbital Processing Production Rule uses the “period” and “boundary” concept just like the Basic Temporal Production Rule. The difference is that for Orbital Processing, the orbit of the spacecraft is taken into account when a PGE or its data are marked as **orbit scheduled**.

When responding to a Production Request for orbit-scheduled processing, PDPS determines the orbit of the satellite via information provided during the SSI&T process. The information (stored in the PDPS database) gives the start time and length of a particular orbit or set of orbits. PDPS extrapolates (or interpolates in the case of an orbit between two orbital periods stored in the database) the start and end times of the PGE that is specified in the Production Request. Data are sought on the basis of the derived start and stop times and the appropriate data granule(s) is/are staged before the PGE is executed.

Orbital path is the path of the satellite over the Earth. It is a number from 0-233 that indicates the region of the Earth covered by a particular orbit. Note that because of the implementation of Orbital Path, there needs to be a mapping between the orbital path calculated by PDPS and the orbital path number expected by the PGEs.

Runtime parameters can be set to values associated with Orbital Processing. The following list of orbital parameters can be placed under runtime parameters:

- Orbit Number.
  - The number of the orbit (starting from zero) and continually increasing.
- Orbital Path Number.
  - The number of the path that maps to the orbit number.
  - The orbital path number is the 0-233 orbital path traversed by the satellite.
- Orbit Number within the Day.
  - The number of the orbit within the given day.
  - It includes any orbit that starts within the given day.
- Granule Number within the Orbit.
  - The number of the granule within a given orbit.
  - It includes any granule that starts within the given orbit.

Figure 11.7.12-1 provides an illustration of the Orbital Processing Production Rule. The PGE in the diagram takes a two-hour input, but is scheduled based on the orbit time and period of the satellite. PDPS uses the data collected at SSI&T to predict the time of the orbit and performs the query to the Science Data Server for the input based on that extrapolated or interpolated orbital time. Granules of input data are allocated to DPRs based on their ability to cover the time period relevant to the DPR.

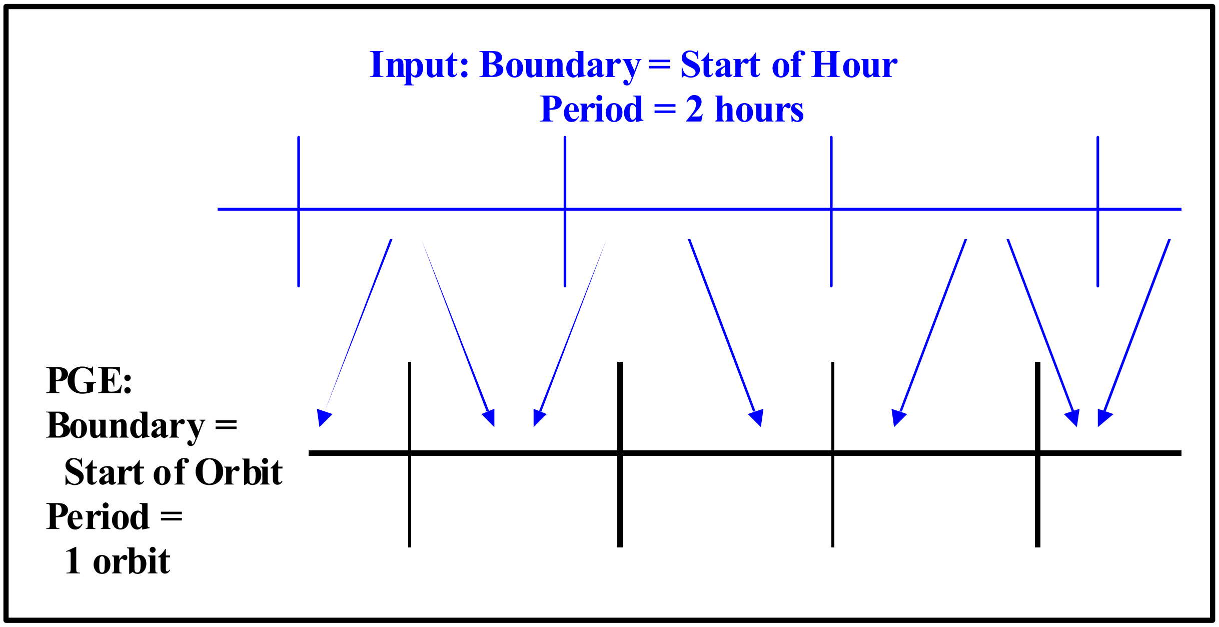


Figure 11.7.12-1. Example of the Orbital Processing Production Rule

In the example shown in Figure 11.7.12-1 the length of an orbit is less than the period of the two-hour input, so sometimes a single granule may cover the input time range of a PGE execution and at other times two granules are required. The production rule would work equally well if the data were of a shorter period (e.g., 1/2 hour) than the orbit of a satellite (e.g., 90 minutes). In such a case three granules would be staged for every execution of the PGE.

The Orbital Processing Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. However, the values specified for the parameters provide orbit information rather than time-period information.

11.7.12.1 PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.
- SCHEDULE_TYPE.
- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The PLATFORM parameter is the name of the platform (satellite) for which the PGE is processing data. Information concerning the orbits of a satellite are stored in the PDPS database. Values that can be assigned to the parameter are subject to the following constraints:

- The string specified can have no more than 25 characters.

- The string specified should match the string specified in the orbit science metadata ODL file. If no matching file is found, an error is reported during SSI&T.

The SCHEDULE_TYPE parameter describes the type of scheduling that is required for the PGE. "Orbit" is the value used for Orbital Processing. As a result, the PGE is scheduled based on the start time and period of the satellite's orbit. Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY must be set correspondingly.

The PROCESSING_PERIOD is the time interval for the data that the PGE processes. Assuming no combination of production rules that would affect the period, data are acquired for the specified PROCESSING_PERIOD and output data are planned for the given period. The value assigned to PROCESSING_PERIOD is of the format "<Period Type>=<Length of Period>". The “Period Type” applicable to the Orbital Processing Production Rule is "ORBITS". For example, "ORBITS=1" would be applied to a PGE that processes data related to one orbit’s worth of data.

The PROCESSING_BOUNDARY is the boundary (starting point in time) of the PGE. It specifies when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction when scheduling the PGE. Consequently, "START_OF_ORBIT" is the acceptable PROCESSING_BOUNDARY for the Orbital Processing Production Rule. It indicates that the PGE processes data related to each satellite orbit. It must be used in conjunction with a PROCESSING_PERIOD that specifies a “Period Type” of “ORBITS”.

11.7.12.2 Orbit Science Metadata ODL File Parameters

The following parameter must be set in the orbit science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.

In addition, the following ODL object is used in defining orbits for the Orbital Processing Production Rule:

- ORBIT_MODEL object.

The value assigned to the PLATFORM parameter in the orbit science metadata ODL file must be exactly the same as that specified for the same parameter in the PGE science metadata ODL file.

The ORBIT_MODEL object is an ODL object that surrounds each orbit definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each orbit that is to be expressly defined: PDPS extrapolates or interpolates orbits that are not specifically defined within the file.

```
OBJECT = ORBIT_MODEL
   CLASS = 1
   ORBIT_NUMBER = 1000
   ORBIT_PATH_NUMBER = 68
   ORBIT_PERIOD = "MINS=98"
```

```
        ORBIT_ START = "09/21/1999 14:50:00"
      END_OBJECT = ORBIT_MODEL
```

The following parameters are set in the ORBIT_MODEL object in order to implement the Orbital Processing Production Rule:

- CLASS.
- ORBIT_NUMBER.
- ORBIT_PATH_NUMBER.
- ORBIT_PERIOD.
- ORBIT_START.

CLASS is a simple counter used to differentiate the different ORBIT_MODEL objects within the file. Each ORBIT_MODEL object needs to have a different CLASS value.

ORBIT_NUMBER is simply the number of the orbit being specified. Each orbit of the satellite has a sequential number associated with it. This is the integer value of the orbit number for the orbit being defined in the ORBIT_MODEL object.

ORBIT_PATH_NUMBER is value of the path for the specified orbit. The orbital path is a number from 0-233 that repeats every 16 days. This is the integer value of the orbital path number for the orbit being defined in the ORBIT_MODEL object.

ORBIT_PERIOD is the length of time it takes for the satellite to complete one orbit. The value assigned to ORBIT_PERIOD has the format "<Period Type>=<Length of Period>" (e.g., "MINS=98"). Note that the "Length of Period" is specified as a positive integer only.

Period Type values for the orbit model science metadata OFL file are:

- "WEEKS"
  - Orbit spans some number of weeks.
  - For example, "WEEKS=2" would be an orbit that takes two weeks to complete.
- "DAYS"
  - Orbit spans some number of days.
  - For example, "DAYS=5" would be an orbit that takes five days to complete.
- "HOURS"
  - Orbit spans some number of hours.
  - For example, "HOURS=4" would be an orbit that takes four hours to complete.
- "MINS"
  - Orbit spans some number of minutes.
  - For example, "MINS=85" would be an orbit that takes eighty-five minutes to complete.
- "SECS"
  - Orbit spans some number of seconds.

    - For example, "SECS=7200" would be an orbit that takes 7200 seconds (two hours) to complete.

ORBIT_START is the start date and time for the orbit defined by the particular ORBIT_MODEL object. Its format is either "MMM DD YYYY HH:MM:SS" or “MM/DD/YYYY HH:MM:SS”.

11.7.13 Production Planning Considerations

During normal operations it is expected that the Production Planner will not have to add PRs to the PDPS database very frequently. The frequency of this activity is, to some extent, determined by the SCF responsible for the science software.

- The PR is a template request to generate a particular data product and results in a production run of the associated SCF-provided PGE.
    - PR specifies a range (temporal, orbit, or tile) over which the data products are to be produced or the PGEs are to be scheduled.
        - PR might request that the data product be produced for only a single day’s data.
        - PR might request that data products be produced for every opportunity of input data for several months, resulting in several hundred jobs being planned and run as the input data become available.
    - Early in a mission the SCF may prefer to request processing for a short time period only (e.g., a week or less).
        - At that time the SCF is gaining an understanding of the on-orbit behavior of the instrument, the resulting data, and the interaction of the science processing software with real data.
        - SCF reviews the quality of the products and notifies the Production Planner of the need for any changes to the PR (e.g., discontinue the PR, change time ranges, or modify input parameters).
    - When the SCF has developed a good understanding of the instrument’s behavior, the team may be comfortable requesting processing for months at a time.
    - DAAC operations may have operational reasons for wanting to issue processing requests for a more limited time period.

The Production Planner has to balance the various considerations when determining whether or not to create or update a PR.

Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs’ data processing needs. The production planning tools are intended to be flexible enough in their design to support the particular planning and scheduling cycles of the operations organization at each DAAC.

Before planning production the Production Planner must coordinate with the Resource Planner to resolve all resource allocation issues. The Resource Planner notifies the Production Planner of the resources available for use in processing. Furthermore, the Production Planner may well have direct access to the Resource Plan.

The Production Planner prepares monthly and weekly production plans. In addition, the Production Planner develops a daily production schedule from the most current weekly plan. However, the first step in the planning process is creating production requests using the Production Request Editor.

This page intentionally left blank.

12. Resource Planning

12.1 Resource Planning Process

The Resource Planning process is the mechanism by which reservations for non-routine ground events are defined and controlled. Such events may include testing, corrective maintenance, preventive maintenance or system upgrades, or any other event that requires DAAC production processing resources. Resource planning defines ground events, which are also used in production planning; thus, resource planning can take place whenever a production plan needs to be created. In general, this will occur on a biweekly basis for 30-day plans, on a weekly basis for ten-day plans, and on a daily basis. However, ground events can be entered at any time. The important point is that it is necessary to be aware of the anticipated processing load and upcoming maintenance events for about the next month.

Resource Planning includes two general types of activities; i.e., Resource Definition and Resource Scheduling. The site M&O Resource Planner uses the Resource Editor GUI within the Planning Subsystem to define ECS resources used in production processing. The Resource Planner and Resource Manager use the Resource Scheduler GUI to schedule non-routine events against ECS resources.

Subsequent sections related to Resource Planning address the following topics:

- Section 12.2 An overview of the process for defining resources and step-by-step procedures for using the Resource Editor.
- Section 12.3 An overview of the process for scheduling resources and step-by-step procedures for using the Resource Scheduler.
- Section 12.4 An overview of the process for tuning system parameters related to Resource Planning and a description of the process for changing configuration parameters.
- Section 12.5 An overview of the process and step-by-step procedures for troubleshooting Resource Planning problems.

12.2 Defining Resources

The Resource Planner uses the Resource Editor GUI to define ECS resources used in production processing in the following terms:

- “Disks.”
- “Virtual computers” (sets of CPUs and associated memory and disks).
- “Strings” (sets of virtual computers).
- “Real computers” (hosts that are composed of one or more virtual computers).
- “AutoSys” (“strings” associated with the production processing software).
- Generic “hardware.”

The following general process is used for defining production resources:

- Determine what production resources are available.
- Determine the distribution of resources among operating modes.
- Define resources for each mode using the Resource Editor GUI.

Each procedure outlined has an **Activity Checklist** table that provides an overview of the task to be completed. The outline of the **Activity Checklist** is as follows:

Column one - ***Order*** shows the order in which tasks could be accomplished.

Column two - ***Role*** lists the Role/Manager/Operator responsible for performing the task.

Column three -***Task*** provides a brief explanation of the task.

Column four - ***Section*** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Column five - **Complete?** is used as a checklist to keep track of which task steps have been completed.

Table 12.2-1, below, provides an Activity Checklist for Defining Resources.

Table 12.2-1. Defining Resources - Activity Checklist

Order	Role	Task	Section	Complete?
1	Resource Planner	Log in to ECS Hosts	(P) 12.2.1	
2	Resource Planner	Launch the Resource Editor	(P) 12.2.2	
3	Resource Planner	Determine Actual Processing Resources	(P) 12.2.3	
4	Resource Planner	Add a Resource	(P) 12.2.4	
5	Resource Planner	Add/Modify a Disk	(P) 12.2.4.1	
6	Resource Planner	Add/Modify a Virtual Computer	(P) 12.2.4.2	
7	Resource Planner	Add/Modify a Real Computer	(P) 12.2.4.3	
8	Resource Planner	Add/Modify a String	(P) 12.2.4.4	
9	Resource Planner	Add/Modify an Autosys Resource	(P) 12.2.4.5	
10	Resource Planner	Add/Modify a Hardware Resource	(P) 12.2.4.6	
11	Resource Planner	Modify a Resource	(P) 12.2.5	
12	Resource Planner	Delete a Resource	(P) 12.2.6	
13	Resource Planner or DAAC Staff	Shut Down Resource Definition Applications	(P) 12.2.7	

12.2.1 Log in to ECS Hosts

Logging in to ECS hosts is accomplished from a UNIX command line prompt. Table 12.2-2 presents (in a condensed format) the steps required to log in to ECS hosts. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedures that follow.

1 At the UNIX command line prompt enter:

setenv DISPLAY <client name>:0.0

- Use either the X terminal/workstation IP address or the machine-name for the client name.
- When using secure shell, the DISPLAY variable is set just once, before logging in to remote hosts. If it were to be reset after logging in to a remote host, the security features would be compromised.

2 In the terminal window (at the command line prompt) start the log-in to the appropriate host by entering:

/tools/bin/ssh <host name>

- The **-l** option can be used with the ssh command to allow logging in to the remote host (or the local host for that matter) with a different user ID. For example, to log in to x0pls02 as user cmops enter:

  /tools/bin/ssh -l cmops x0pls02

- Depending on the set-up it may or may not be necessary to include the path (i.e., /tools/bin/) with the ssh command. Using ssh alone is often adequate. For example:

  ssh x0pls02

  - or -

  ssh -l cmops x0pls02

- Examples of Planning/Management Workstation host names include **e0pls03**, **g0pls01,** and **l0pls02.**
- Examples of Science Processor host names include **e0spg11**, **g0spg11,** and **l0spg11**.
- Examples of Queuing Server host names include **e0sps04**, **g0sps06**, and **l0sps03**.
- Examples of Sun internal server host names include **e0acs06**, **g0acs06**, and **l0acs06**.
- Examples of Access/Process Coordinators (APC) Server host names include **e0acg11**, **g0acg01**, and **l0acg02**.
- Examples of Ingest Server host names include **e0icg11, g0icg01**, and **l0acg02**.
- Examples of Sun external server host names include **e0ins01, g0ins01**, and **l0ins01**.

- If you receive the message, “Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?” enter **yes** (“y” alone will not work).
- If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter *passphrase* for RSA key '<*user@localhost*>'** appears; continue with Step 3.
- If you have not previously set up a secure shell passphrase, go to Step 4.

3 If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, enter:

<passphrase>

- If a command line prompt is displayed, log-in is complete.
- If the passphrase is unknown, press **Return/Enter**, which should cause a **<user@remotehost>’s password:** prompt to appear (after the second or third try if not after the first one), then go to Step 4.
- If the passphrase is entered improperly, a **<user@remotehost>’s password:** prompt should appear (after the second or third try if not after the first one); go to Step 4.

4 If a prompt for **<*user@remotehost*>’s password:** appears, enter:

<password>

- A command line prompt is displayed.
- Log-in is complete.

Table 12.2-2. Log in to ECS Hosts - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**setenv DISPLAY <client name>:0.0**	**enter text, press Enter**
2	**/tools/bin/ssh <host name>** (as applicable)	**enter text, press Enter**
3	**<passphrase>** (if applicable)	**enter text, press Enter**
4	**<password>** (if applicable)	**enter text, press Enter**

12.2.2 Launch the Resource Editor

The Resource Editor is invoked from a UNIX command line prompt. Table 12.2-3 presents (in a condensed format) the steps required to launch the **Resource Editor** GUI. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedures that follow.

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03**, **g0pls01,** and **l0pls02.**
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

2 In the terminal window, at the command line, enter:

cd /usr/ecs/<MODE>/CUSTOM/utilities

- **<MODE>** is current mode of operation.
  - TS1 - Science Software Integration and Test (SSI&T)
  - TS2 - New Version Checkout
  - OPS - Normal Operations
- "utilities" is the directory containing the Planning Subsystem start-up scripts.

3 Set the application environment variables by entering:

setenv ECS_HOME /usr/ecs/

- Application home environment is entered.
- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to set it manually.

4 Start the Resource Planning background processes by entering:

EcPlRpAllStart <MODE> <application_id>

- The Resource Planning background processes are launched.
- The **application_id** or **MSGSRV_ID** is a number from 1 to 5. It identifies the message service in use so messages can be directed to the proper message handler GUI. Consequently, it is a good idea to use the same application_id consistently during a resource planning session.

5 Start the **Resource Editor** GUI by entering:

EcPlRpReStart <MODE> <application_id>

- The **Resource Editor** GUI is launched.
- The **Resource Editor** GUI displays a list of defined resources and a series of buttons that enable the following operations:
  - **New...** Add a resource definition. (Section 12.2.4)
  - **Modify...** Edit or review the details of an existing resource definition. (Section 12.2.5)
  - **Delete** Delete a resource definition. (Section 12.2.6)
  - **Fetch Baseline** [not used]
  - **Load Baseline** [not used]

Table 12.2-3. Launch the Resource Editor - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	UNIX window (Planning/Management Workstation)	**single-click** or use procedure in Section 12.2.1
2	**cd /usr/ecs/<MODE>/CUSTOM/utilities**	**enter text, press Enter**
3	Set the environment variables	**enter text, press Enter**
4	**EcPlRpAllStart <MODE> <application_id>**	**enter text, press Enter**
5	**EcPlRpReStart <MODE> <application_id>**	**enter text, press Enter**

12.2.3 Determine Actual Processing Resources

The Resource Editor allows the authorized operator to define resources in the following categories:

- **Disks:** Disk partitions that are associated with and provide temporary data storage for the input and output files used in processing.
- **Virtual Computers:** Virtual computers composed of CPUs, random-access memory (RAM), and associated-disk(s). The CPUs and RAM specified for a virtual computer are components of the real computer from which the virtual computer is derived.
- **Strings:** Sets of one or more virtual computers. Strings are associated with the processing software (AutoSys). A dual science processor configuration can be defined by specifying strings containing virtual computers derived from different real computers.
- **Real Computers:** Physical computing devices (hosts), each of which contains one or more CPUs. Each science processor host ("real" computer) is divided into one or more virtual computers by allocating CPUs and RAM from the real computer to the virtual computer(s).
- **AutoSys:** Identifies the string(s) of virtual computers used by the production processing software.
- **Hardware:** Any type of equipment that is not defined as a computer or disk may be defined as "hardware."

The ECS Operational Readiness Plan for Release 2.0 (603-CD-003-001) specifies that initially disk partitions at the DAACs are to be split among the operating modes as follows:

- OPS – 60%.
- TS1 - 20%.
- TS2 - 20%.

However, it may be advantageous to reserve some nominal percentage of the disk (e.g., two to five percent) as a safety buffer. In any case, it is critical to ensure that the sum of the disk space assigned to the various modes is no more than the total disk space available.

Although the ECS Operational Readiness Plan does not specifically mention allocating resources other than disk partitions, CPUs and RAM need to be allocated among modes in the same manner. However, it is not necessary to be exact with the CPU count or RAM amount.

- There is no one-to-one mapping of CPU allocation with actual CPUs on the science processor.
- Actual CPU usage during processing is limited by the operating system (OS).
  - If ten CPUs have been specified for a particular mode, only ten Data Processing Requests (DPRs) can be running the Execute job at a given time.
  - What is really being defined is the maximum number of DPRs that will execute at a given time.
- It is important to monitor the load on each science processor.
  - CPUs can be over-allocated or under-allocated as necessary to get the most out of the CPUs on each science processor.
  - If monitoring indicates that the processor is underused when OPS mode is at full processing capacity, the number of CPUs allocated to OPS mode could probably be increased.
  - If the science processor is at full capacity when OPS mode is at full processing capacity and it is suspected that the processor may be overworked, the number of CPUs allocated to OPS mode should be reduced.
- Random-access memory (RAM) is subject to the same considerations as CPUs.
  - RAM can be over-allocated or under-allocated as necessary to get the most out of the memory on each science processor.
- The OS takes care of true CPU and RAM allocation.

Before data processing resources can be defined, it is necessary to know what resources are actually available at the DAAC. Some resources are defined in terms of other resources; for example, a string is defined as one or more virtual computers. However, it is generally necessary to have the following types of information available in order to define processing resources:

- Host names [“real computers”].
- Number of processors [CPUs] available on each host.
- Operating System (OS) for each host.
- Memory [RAM] on each host.
- Total disk space.
- AutoSys instance(s) at the DAAC.

Table 12.2-4 presents (in a condensed format) the steps required to determine the actual processing resources to be defined. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

NOTE: The procedure to determine the actual processing resources to be defined starts with the assumption that the DISPLAY environment variable has been set (Refer to Section 12.2.2).

1 Access a terminal window logged in to the applicable Science Processor host.

- Examples of Science Processor host names include **e0spg11**, **g0spg11,** and **l0spg11**.
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

2 To access the mount point enter:

```
cd  /usr/ecs/<MODE>/CUSTOM/pdps/<processor>/data/DpPrRm/<processor>_disk
```

- Change directory to the disk mount point (e.g., /usr/ecs/OPS/CUSTOM/pdps/e0spg11/data/DpPrRm/e0spg11_disk).

3 To determine disk size and usage enter:

```
df  -k  .
```

(being sure to include the dot)

- Information concerning disk size and use is displayed; for example:

```
Filesystem          Type  kbytes     use     avail  %use Mounted on
/dev/dsk/xlv/vgo1       xfs 413394688 164646048 248748640  40  /vol1
```

- In the preceding example the total disk space is 413,394,688 kilobytes or 413,394.69 megabytes (413 gigabytes).

4 To obtain Information concerning the number of CPUs and amount of RAM (memory) enter:

```
hinv
```

- The hinv command is available on Silicon Graphics, Inc. (SGI) hosts only.
- Information concerning CPUs and RAM (memory) is displayed; for example (not all rows are shown):

```
Processor 0: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.6
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 1: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.6
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 2: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.6
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 3: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.6
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 4: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
```

```
Processor 5: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 6: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 7: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 8: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 9: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 10: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 11: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 12: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 13: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 14: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Processor 15: 194 MHZ IP25
CPU: MIPS R10000 Processor Chip Revision: 2.5
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
Secondary unified instruction/data cache size: 1 Mbyte
Data cache size: 32 Kbytes
Instruction cache size: 32 Kbytes
Main memory size: 2048 Mbytes, 8-way interleaved
[…]
```

- In the example the science processor has 16 CPUs (Processor 0 – Processor 15) and 2048 megabytes of RAM.

5 Repeat Steps 1 through 4 for all other science processors (if any).

NOTE: Steps 6 through 12 describe the use of the Netscape browser to determine certain types of information concerning computer resources (including the number of CPUs and amount of RAM), which can be determined using the **hinv** command as described in Step 4. However, the "as-built" file accessed using the Netscape browser lists the necessary operating system information in addition to CPU and RAM data. The advantage of the **hinv** command is that it provides real-time data and is reliably up to date. The advantage of the "as-built" file accessed using the Netscape browser is that it provides operating system data that is not available using the **hinv** command.

6 To launch the Netscape web browser enter:

netscape &

- It may be necessary to change directories before launching the Netscape web browser (e.g., cd /tools/bin/netscape3.01).
- The Netscape web browser is displayed.

7 In the browser's **Location** (**Go To**) field enter the following address:

http://cmdm.east.hitc.com/baseline

- The ECS Baseline Information System web page is displayed.

8 **Single-click** on the **ECS Configuration** link.

- A table of files is displayed.

9 **Single-click** on the **Asbuilts** link for the relevant DAAC.

- A list of files is displayed.

10 **Single-click** on the file name corresponding to the desired host.

- For example, the file name for host x0spg01 would be x0spg01.asbuilt.html.
- A report containing the following types of information (among other items) is displayed:
  - **Host Name** ["real computer"].
  - **Processors** [CPUs].
  - **Operating System.**
  - **Memory** [RAM].
  - **Interrogation Date** (useful in determining how up-to-date the information is).

11 **Single-click** on the browser **Back** button.

- The list of "as-built" files is displayed.

12 Repeat Steps 10 and 11 for all other science processors (if any).

13 Exit from the Netscape browser when the necessary information has been acquired by executing the following menu path:

File → Exit

- The Netscape browser disappears.

14 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04**, **g0sps06**, and **l0sps03**.
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

15 To gain access to the directory containing the AutoSys configuration files enter:

cd /usr/ecs/<MODE>/COTS/<autotree>/autouser

- Change directory to the directory (e.g., **/usr/ecs/<MODE>/COTS/autotreeb/autouser**, **/usr/ecs/<MODE>/COTS/autotree/autouser**, **/data1/SHARED/COTS/autotree/autouser**) containing the set-up files (e.g., **FMR.autosys.csh.g0sps06**) and the AutoSys configuration files (e.g., **config.FMR**).
- The particular path to be typed may vary from site to site.
- The AutoSys instance at the DAAC is identified by three capital letters appended to the beginning of the set-up files and the end of the configuration file.
  - Typically, AutoSys instances at the DAACs are identified as **FMR**.
- It is possible to have multiple AutoSys instances installed at a DAAC.

Table 12.2-4. Determine Actual Processing Resources - Quick-Step Procedures (1 of 2)

Step	What to Enter or Select	Action to Take
1	UNIX window (Science Processor)	**single-click** or use procedure in Section 12.2.1
2	**cd /usr/ecs/<MODE>/CUSTOM/pdps/<processor> /data/DpPrRm/<processor>_disk**	**enter text, press Enter**
3	**df -k .** (being sure to include the dot)	**enter text, press Enter**
4	Observe the disk capacity	**read text**
5	**hinv**	**enter text, press Enter**
6	Observe the number of CPUs and total memory (RAM)	**read text**
7	Repeat Steps 1 through 6 for all other science processors (if any)	

Table 12.2-4. Determine Actual Processing Resources - Quick-Step Procedures (2 of 2)

Step	What to Enter or Select	Action to Take
8	Launch Netscape	**enter text, press Enter**
9	**http://pete.hitc.com/baseline**	**enter text, press Enter**
10	**ECS Configuration** link	**single-click**
11	**Asbuilts** link (for the relevant DAAC)	**single-click**
12	**<file name>** (corresponding to the desired host)	**single-click**
13	Observe the number of CPUs, total memory (RAM), and Operating System identification	**read text**
14	**Back** button	**single-click**
15	Repeat Steps 12 through 14 for all other science processors (if any)	
16	**File** → **Exit** (to exit from Netscape)	**single-click**
17	UNIX window (Queuing Server)	**single-click** or use procedure in Section 12.2.1
18	**cd /usr/ecs/<MODE>/COTS/<autotree>/autouser**	**enter text, press Enter**
19	Observe the identification of the AutoSys instance	**read text**

12.2.4 Add a Resource

These procedures address adding such resources as computers, disk partitions, strings, and generic hardware resources to the resource planning list. The ECS Operational Readiness Plan for Release 2.0 (603-CD-003-001) specifies that initially disk partitions at the DAACs are to be split among the operating modes as follows:

- OPS – 60%.
- TS1 - 20%.
- TS-2 - 20%.

However, it may be advantageous to reserve some nominal percentage of the disk (e.g., two to five percent) as a safety buffer. In any case, it is critical to ensure that the sum of the disk space assigned to the various modes is no more than the total disk space available.

Table 12.2-5 presents (in a condensed format) the steps required to add a resource. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Editor** GUI (refer to Section 12.2.2).

- The **Resource Editor** GUI is displayed.

2 From the **Resource Editor** GUI, select the type of resource to be added from the list on the **Resource Type** button.

3 Either **single-click** on the **New…** button or execute the following menu path:

File → New

4 Define the resource as specified in the corresponding procedure section.

- Refer to the specified section for defining the desired type(s) of resources:
  - **Disk** – Section 12.2.4.1.
  - **Virtual Computer** – Section 12.2.4.2.
  - **Real Computer** – Section 12.2.4.3.
  - **String** – Section 12.2.4.4.
  - **Autosys** – Section 12.2.4.5.
  - **Hardware** – Section 12.2.4.6.
- Resources should generally be added in the preceding order (due to dependencies among resources).

5 After the data have been entered, **single-click** on one of the following buttons:

- **Save** to save **and** exit.
- **Cancel** to exit without saving.

Table 12.2-5. Add a Resource - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Editor** GUI (if necessary)	Use procedure in Section 12.2.2
2	**Resource Type** option button	**single-click**
3	**New…** button	**single-click**
4	Make entries in the necessary fields	Use procedures in Sections 12.2.4.1 through 12.2.4.6
5	**Save** button	**single-click**

12.2.4.1 Add/Modify a Disk

1 Enter the relevant information in the following fields on the **Disk Details** GUI:

- **Resource Name** Operator-defined name for the resource. (required)
  - Example: e0spg11_disk_OPS.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the **Activity** field and then clicking on one of the available options.
- **Partition Size** The size of the disk partition, in **megabytes**. (required)
  - Although the label on the GUI implies that partition size should be entered in "blocks," the label is erroneous. Enter the partition size in megabytes.

- **Block Size** Block size in bytes (always 1024) used for the disk. (required)
- **Comments** Operator comments on the resource.

2 After the data have been entered, **single-click** on one of the following buttons:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.4.2 Add/Modify a Virtual Computer

1 Enter the relevant information in the following fields on the **Virtual Computer Details** GUI:
- **Resource Name** Operator-defined name for the virtual computer. (required)
  - Example: e0spg11_vc_OPS.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the Activity field and then clicking on one of the available options.
- **Number of CPUs** Number of CPUs within the virtual computer. (required)
- **Total RAM** The total memory for the virtual computer in megabytes. (required)
- **Operating System** The operating system name/version for the computer. (required)
- **Disks** A list of the disks previously defined for that site. This list of disks from which to select is used when a disk is associated (or disassociated) with the computer. After items are highlighted, arrow buttons will move items from this list to **Associated Disks** or from the list of **Associated Disks** to the **Disk** list.
- **Associated Disks** Disks in this list are associated with the computer.
- **Comments** Operator comments on the resource.

2 After the data have been entered, **single-click** on one of the following buttons:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.4.3 Add/Modify a Real Computer

1 Enter the relevant information in the following fields on the **Real Computer Details** GUI:
- **Resource Name** Operator-defined name for the real resource. (required)
  - Example: e0spg11.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the Activity field and then clicking on one of the available options.
- **Computers** A list of the virtual computers previously defined for that site. This list of virtual computers from which to select is used when a virtual computer is associated (or disassociated) with the real computer. After items are highlighted,

arrow buttons will move items from this list to **Associated Computers** or from the list of **Associated Computers** to the **Computers** list.
- **Associated Computers** Virtual computers in this list are associated with the real computer.
- **Comments** Operator comments on the resource.

2 After data have been entered, **single-click** on the appropriate button from the following selections:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.4.4 Add/Modify a String

1 Enter the relevant information in the following fields on the **String Details** GUI:
- **Resource Name** Operator-defined name for the resource. (required)
  - Example: e0spg11_string_OPS.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the Activity field and then clicking on one of the available options.
- **Computers** A list of virtual computers previously defined for that site. This list of computers from which to select is used when a computer is associated (or disassociated) with the string. After items are highlighted, arrow buttons will move items from this list to **Associated Computers** or from the list of **Associated Computers** to the **Computer** list.
- **Associated Computers** Virtual computers in this list are associated with the string.
- **Comments** Operator comments on the resource.

2 After data have been entered, **single-click** on the appropriate button from the following selections:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.4.5 Add/Modify an AutoSys Resource

1 Enter the relevant information in the following fields on the **Autosys Details** GUI:
- **Resource Name** Operator-defined name for the AutoSys resource. (required)
  - Example: FMR.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the Activity field and then clicking on one of the available options.
- **Strings** A list of the strings previously defined for that site. This list of strings from which to select is used when a string is associated (or disassociated) with the AutoSys resource. After items are highlighted, arrow buttons will move items from this list to **Associated Strings** or from the list of **Associated Strings** to the **Strings** list.
- **Associated Strings** Strings in this list are associated with the AutoSys resource.

- **Comments** Operator comments on the resource.

2 After data have been entered, **single-click** on the appropriate button from the following selections:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.4.6 Add/Modify a Hardware Resource

1 Enter the relevant information in the following fields on the **Hardware Resource Details** GUI:
- **Resource Name** Operator-defined name for the resource. (required)
  - Example: e0spg11_cdrom_OPS.
- **Activity** System-generated default activity; can be changed by clicking on the bar in the Activity field and then clicking on one of the available options.
- **Comments** Operator comments on the resource.

2 After data have been entered, **single-click** on the appropriate button from the following selections:
- **Save** to save **and** exit**.**
- **Cancel** to exit **without** saving.

12.2.5 Modify a Resource

Table 12.2-6 presents (in a condensed format) the steps required to modify a resource. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Editor** GUI (refer to Section 12.2.2).
- The **Resource Editor** GUI is displayed.

2 From the list of resources displayed on the **Resource Editor** GUI, **single-click** on the resource to be modified.

3 **Single-click** on the **Modify…** button to access the appropriate Details GUI.

4 Make the modifications.
- For field descriptions, refer to Sections 12.2.4.1 through 12.2.4.6.

5 After data have been entered, **single-click** on the appropriate button from the following selections:
- **Save** to save **and** exit**.**
- Cancel to exit without saving.

Table 12.2-6. Modify a Resource - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Editor** GUI (if necessary)	Use procedure in Section 12.2.2
2	Select the resource to be modified	**single-click**
3	**Modify…** button	**single-click**
4	Make the modifications	Use procedures in Sections 12.2.4.1 through 12.2.4.6
5	**Save** button	**single-click**

12.2.6 Delete a Resource

Table 12.2-7 presents (in a condensed format) the steps required to delete a resource. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Editor** GUI (refer to Section 12.2.2).

- The **Resource Editor** GUI is displayed.

2 From the list of resources displayed on the **Resource Editor** GUI, **single-click** on the resource to be deleted.

3 **Single-click** on the **Delete** button.

- A dialogue box pops up to verify whether the resource is really to be deleted.

4 **Single-click** on one of the following buttons as appropriate:

- **OK** to remove the resource from the list and from the PDPS database **and** exit**.**
- **Cancel** to exit **without** deleting the resource.

Table 12.2-7. Delete a Resource - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Editor** GUI (if necessary)	Use procedure in Section 12.2.2
2	**<resource>** (to be deleted)	**single-click**
3	**Delete** button	**single-click**
4	**OK** button	**single-click**

12.2.7 Shut Down Resource Definition Applications

When resource definition activities have been completed, the Message Handler, System Name Server, and Resource Model should be shut down to eliminate unneeded processes and allow other operators to gain access to the resource planning applications. If any of the three processes remains active, it is likely to interfere with subsequent attempts to launch resource planning applications.

Shutting down resource definition applications starts with the assumption that the **Resource Editor** GUI is currently being displayed.

Table 12.2-8 presents (in a condensed format) the steps required to shut down resource definition applications. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 To exit from the **Resource Editor** GUI when resource planning activities have been completed execute the following menu path:

File → Exit

- The **Resource Editor** GUI disappears.

2 After quitting the **Resource Editor** GUI **single-click** in the UNIX window used to start the resource definition applications.

3 Shut down the Message Handler, System Name Server, and Resource Model by entering:

EcPlRpSlayAll <MODE> <application_id>

- The **Message Handler** GUI disappears.

4 To obtain a list of active processes in the specified mode enter:

ps -ef | grep <MODE>

- A list of active processes in the specified mode is displayed.
- If an error message is received when **ps -ef | grep <MODE>** is entered, enter:

  ps -auxwww | grep <MODE>

5 Examine the list of processes running in the specified mode to determine whether the Message Handler, System Name Server, and Resource Model processes have actually been shut down.

- None of the following processes should be active:
  - EcPlRpRe
  - EcPlRpSi
  - EcPlRpTl
  - EcPlMsh
  - EcPlSns
  - EcPlRpRm

6 If any of the specified processes [especially the Message Handler, System Name Server, and/or Resource Model process(es)] is/are still active, terminate the active process(es)by entering:

kill -15 <process ID1> [<process ID2>] [<process ID3>] […]

7 Repeat Steps 4 through 6 as necessary.

Table 12.2-8. Shut Down Resource Definition Applications - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**File** → **Exit** (to quit the **Resource Editor** GUI)	**single-click**
2	**OK** button	**single-click**
3	UNIX window (Planning/Management Workstation)	**single-click**
4	**EcPlRpSlayAll <MODE>** **<application_id>**	**enter text, press Enter**
5	**ps -ef | grep <MODE>**	**enter text, press Enter**
6	Identify resource planning process(es) that has (have) not shut down	**read text**
7	**kill -15 <process ID1> [<process ID2>] [<process ID3>] […]** if necessary	**enter text, press Enter**

12.3 Scheduling Resources

The Resource Planner and Resource Manager are both involved in resource scheduling using the Resource Scheduler. The Production Planner and Production Monitor are involved in the implementation of ground events.

- Resource Planner processes resource reservation requests for ground events.
- Resource Manager commits resource reservations.
- Production Planner sends committed resource reservations (ground events) to Data Processing via the Planning Workbench.
- Production Monitor monitors execution of ground events in processing.

The following process is used for generating and implementing resource reservations (ground events):

- Personnel who have a need for Planning Subsystem or Data Processing Subsystem resources submit requests for time on specified resources to accomplish the non-routine activities that they plan to undertake.
  - Depending on DAAC policy, many personnel may have access to the resource planning applications for creating resource reservation requests.
  - Alternatively, personnel may have to contact the Resource Planner to have resource reservation requests entered for them.

- The Resource Planner reviews requests for resource reservations to determine if the requests are valid.
  - Request information includes a description of the activity, the resources required, the time period(s) for using the requested resource(s), comments explaining the variance from normal use.
  - Resource Planner may decide to forward the request to a “sponsor” for validation.
  - A sponsor is someone who evaluates a resource reservation request based on expertise that is particularly relevant to the resource reservation request.
- If the Resource Planner or sponsor determines that the request to reserve the resource is valid, the Resource Planner “approves” it along with all other requests that have been validated.
  - The set of all validated resource reservation requests is considered a draft Resource Plan.
- The scheduling software identifies conflicts (if any) in the draft Resource Plan and alerts the Resource Planner to the problem(s).
- If possible, the Resource Planner resolves all conflicts before presenting the proposed plan to the Resource Manager to have the resources committed.
  - However, during the conflict-resolution process the Resource Planner may have to consult with resource requesters and the Resource Manager to ensure that the reserved resources will not have adverse effects on the DAAC’s high-priority events.
- When the Resource Planner has achieved a conflict-free plan, it is presented to the Resource Manager to be implemented.
- The Resource Manager "commits" the resource plan, which signals the Planning Subsystem that the plan can be implemented.
  - Committing a plan actually involves committing all of the individual approved resource reservation requests that collectively make up the plan.
- All committed resource reservations are automatically included in the next production plan to be activated through the Planning Workbench and are subsequently sent to Data Processing.
  - Resource reservations/ground events are not sent to data processing until they have been included in a production plan.
  - Refer to Section 13, Production Planning, to see how production plans are created and activated.
- In Data Processing a ground event job for each resource reservation is sent to the specified resource(s) at the indicated start time.
  - If a data processing job is already using the specified resource(s) at the ground event’s scheduled start time, the data processing job runs to completion before releasing the resource(s) to the ground event job.

Table 12.3-1, below, provides an Activity Checklist of Resource Scheduling activities.

Table 12.3-1. Resource Scheduling - Activity Checklist

Order	Role	Task	Section	Complete?
1	Resource Planner or DAAC Staff	Launch the Resource Scheduler	(P) 12.3.1	
2	Resource Planner or DAAC Staff	Create a Resource Reservation Request	(P) 12.3.2	
3	Resource Planner or DAAC Staff	Edit a Resource Reservation Request	(P) 12.3.3	
4	Resource Planner or Sponsor	Validate or Reject a Resource Reservation Request	(P) 12.3.4	
5	Resource Planner	Approve a Resource Reservation Request	(P) 12.3.5	
6	Resource Manager/Resource Planner	Commit Resource Reservation Requests	(P) 12.3.6	
7	Resource Planner	Review the Resource Timeline	(P) 12.3.7	
8	Resource Planner	Delete a Resource Reservation Request	(P) 12.3.8	
9	Resource Planner or DAAC Staff	Shut Down the Resource Scheduler	(P) 12.3.9	

12.3.1 Launch the Resource Scheduler

The **Resource Scheduler** is invoked from a UNIX command line prompt. Table 12.3-2 presents (in a condensed format) the steps required to launch the **Resource Scheduler**. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03**, **g0pls01,** and **l0pls02.**
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

2 In the terminal window, at the command line, enter:

cd /usr/ecs/<MODE>/CUSTOM/utilities

- **<MODE>** is current mode of operation.
  - TS1 - Science Software Integration and Test (SSI&T)
  - TS2 - New Version Checkout
  - OPS - Normal Operations
- “utilities” is the directory containing the Planning Subsystem start-up scripts.

3 Set the application environment variables by entering:

setenv ECS_HOME /usr/ecs/

- Application home environment is entered.
- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to set it manually.

4 Start the Resource Planning background processes by entering:

EcPlRpAllStart <MODE> <application_id>

- The Resource Planning background processes are launched.
- The **application_id** or **MSGSRV_ID** is a number from 1 to 5. It identifies the message service in use so messages can be directed to the proper message handler GUI. Consequently, it is a good idea to use the same application_id consistently during a resource planning session.

5 Start the **Resource Scheduler** GUI by entering:

EcPlRpSiStart <MODE> <application_id>

- The Resource Scheduler is launched.
- The **Resource Scheduler** Graphical User Interface (GUI) is accessed. The GUI displays the Resource Reservation List, activity type, and a series of buttons that enable the following operations:
  - **New...** Create a resource reservation request. (Section 12.3.2)
  - **Modify...** Edit or review the details of an existing resource reservation request. (Section 12.3.3)
  - **Approve** Used to indicate that the resource reservation request(s) has (have) been validated and a draft resource plan can be created. Clicking on this button causes the Planning Subsystem to determine whether there are conflicts between this resource reservation and other reservations. The Planning Subsystem detects conflicts and reports them to the operator. (Section 12.3.5)
  - **Commit globally** Commit all "approved" resource reservations; at this point the ground events will be accessible by the production planning software (Section 12.3.6); however, resource reservations/ground events are not sent to data processing until they have been included in an activated production plan.
  - **Time Line** Display a timeline-oriented view of the resource plan. (Section 12.3.7)
  - **Report** The **Report** option is disabled. The reports have been deleted from the system requirements.

Table 12.3-2. Launch the Resource Scheduler - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	UNIX window (Planning/Management Workstation)	**single-click** or use procedure in Section 12.2.1
2	**cd /usr/ecs/<MODE>/CUSTOM/utilities**	**enter text, press Enter**
3	Set the environment variables if necessary	**enter text, press Enter**
4	**EcPlRpAllStart <MODE>** <application_id>	**enter text, press Enter**
5	**EcPlRpSiStart <MODE>** <application_id>	**enter text, press Enter**

12.3.2 Create a Resource Reservation Request

Table 12.3-3 presents (in a condensed format) the steps required to create a Resource Reservation Request. If you are already familiar with the procedures, you may prefer to use this quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures.

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).

- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI, **single-click** on the **New...** button to access the **Resource Reservation Request Edit/Definition** GUI.

3 Enter resource request identification information into the displayed fields. Press **Tab** to move from field to field. **NOTE:** Data that is system-generated is identified.

- **Request Name** Operator-provided name for the resource request. (required)
- **Edited Date** System-generated date of request entry.
- **Originator** Operator-provided name of the authorized user preparing the resource request.
- **Sponsor** Operator -provided name of the individual who is to review and validate the Resource Request (the subject-matter-expert). (required)

4 Enter resource scheduling information into the displayed fields. Press **Tab** to move from field to field.

- **Activity Type** Operator-provided description of the type of activity; selected by the operator from a selection list of valid options. (required)
- **Priority** Operator-provided priority for the activity. Use the slider to select the appropriate priority on a scale from 0 to 100. One (1) denotes the highest priority and 100 designates the lowest.
- **Description** Operator-provided description of the activity for which the resource is required. (required)
- **Resource...** See Section 12.3.2.1, below. (required)
- **Interval...** Not applicable to new resource reservation requests; may be applicable when editing a resource reservation request. See Section 12.3.3.1, below.

5 Enter duration information into the displayed fields to define the period over which the resource is required. Press **Tab** to move from field to field.

- **Start Date** Operator-provided start date of the resource request period. Enter in <**MM/DD/YYYY**> format. (required)
- **Start Time** Operator-provided start time of the resource request. Enter in <**hh:mm:ss**> format. (required)
  - **Start Time** must be later than the time when the resource reservation request will be saved; otherwise, it will not be possible to save the request.
- **Stop Date** Operator-provided stop date of the resource request period. Enter in <**MM/DD/YYYY**> format. If a reservation is to be repeated over some **frequency** (see below), the stop date specifies the end date in the date range of the reservation request. (required)
- **Stop Time** Operator-provided stop time of the resource request. Enter in <**hh:mm:ss**> format. (required)
- **Frequency** See Section 12.3.2.2, below.

6 Enter comments concerning the resource reservation request in the **Comment** field.

7 After data have been entered, **single-click** on the appropriate button from the following selections:

- **Save** to save data.
  - The resource reservation must be "saved" prior to validating or rejecting. After the request has been saved, it can then be **validated** or **rejected**.
  - The selections **Validated** and **Rejected** are further discussed in Section 12.3.3.
- **Clear** to clear entries. Once cleared, the entries are deleted from the system.
- **Cancel** to exit screen without saving the request.

Table 12.3-3. Create a Resource Reservation - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**New…** button	**single-click**
3	**<resource identification information>**	**enter text, press tab**
4	**<resource scheduling information>**	**enter text, press tab**
5	**<duration information>**	**enter text, press tab**
6	**<comments>** (optional)	**enter text, press tab**
7	**Save** button	**single-click**

12.3.2.1 Selecting Resources...

Clicking on the **Resource…** button accesses a **Resources Selection** screen. The **Request Name** is blank and is to remain empty when creating a new resource reservation request. This screen provides a pair of lists: **Resources** and **Selected Resources**. The **Resources** list itemizes available resources. The **Selected Resources** list itemizes those resources that have been selected for incorporation into the resource reservation. The operator selects the desired resource(s) and, using the arrow buttons, moves the resource(s) from one list to the other list.

1 **Single-click** on your selections in the list and **single-click** on the desired arrow to move resources between the **Resources** and **Selected Resources** lists.

2 **Single-click** on one of the following buttons as appropriate:
- **OK** to save the selections and exit the screen.
- **Cancel** to exit the screen **without** saving changes.

12.3.2.2 Selecting Frequency

The **Frequency** option button provides the mechanism that allows the operator to specify whether the resource reservation request describes a one-time event or a recurring event. Clicking on **Frequency** allows the operator to specify options for periodic resource requests; that is, to specify the frequency of occurrence of a repeating resource need. Several options for expressing the frequency are available in the **Frequency** selection list box combined with a text field that provides a qualifier (i.e., number of days) for the **Every_?_days** selection only. The frequency specified defaults to **Once** to indicate that the resource need covers the entire time period covered by 'Start Time' and 'Stop Time.' Other options are identified in Table 12.3-4. The dates generated are inserted in the **Selected Intervals** list box, described in Section 12.3.2.3, below.

Table 12.3-4. Frequency Qualifiers (1 of 2)

Frequency	Text Qualifier:	Result:
Once	--	The default. Resource reservation covering the period from the start time and stop time for the start date specified.
Daily	--	Resource reservation for every day, between the start date and end date, for the start time and end time specified.
Weekly	--	Resource reservation for every week occurring on the day specified by the start date, repeated until the end date as specified.
Every_2_weeks	--	Resource reservation occurring biweekly on the day specified by the start date, repeated until the end date as specified.
Monthly	--	Resource reservation for every month on the start day of the month, repeated until the end date as specified.
Mon_thru_Fri	--	Resource reservation for every Monday through Friday, between the start date and end date, for the start time and end time specified.

Table 12.3-4. Frequency Qualifiers (2 of 2)

Frequency	Text Qualifier:	Result:
Mon_Wed_Fri	--	Resource reservation for every Monday, Wednesday, and Friday, between the start date and end date, for the start time and end time specified.
Tues_Thurs	--	Resource reservation for every Tuesday and Thursday, between the start date and end date, for the start time and end time specified.
Every_?_days	*n*	Resource reservation for every *n* days, between the start date and end date, for the start time and end time specified.
Weekend	--	Resource reservation for every Saturday and Sunday, between the start date and end date, for the start time and end time specified.

12.3.3 Edit a Resource Reservation Request

Table 12.3-5 presents (in a condensed format) the steps required to edit a Resource Reservation Request. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).
- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI, **single-click** on the resource reservation request to be modified.

3 **Single-click** on the **Modify…** button to access the **Resource Reservation Request Edit/Definition** GUI.

4 Make the modifications to affected fields. (See Section 12.3.2, above.)
- **Interval...** is applicable when editing a resource reservation request if certain intervals are to be excluded from the resource reservation. See Section 12.3.3.1, below.

5 If appropriate at this time, **single-click** on either **Validated** or **Rejected**.
- **Validated** indicates that the reservation request is complete and 'makes sense'; that is, the request includes the appropriate resources consistent with the type of activity that is being proposed.
- **Rejected** indicates that the reservation request is rejected.
- At this time, the **Comment** field may also be updated.
- The **Status** field contains the status of the reservation request.
  – Status is system-generated based on operator-input in other fields.

6 After data is entered, **single-click** on the appropriate button(s):

- **Save** to save data and exit screen.
- **Clear** to clear entries. Once cleared, the entries are deleted from the system.
- **Cancel** to exit screen.

Table 12.3-5. Edit a Resource Reservation Request - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**<resource reservation request>** (to be modified)	**single-click**
3	**Modify…** button	**single-click**
4	Make modifications to affected fields	Use procedure in Section 12.3.2
5	**Validated** button or **Rejected** button if applicable	**single-click**
6	**Save** button	**single-click**

12.3.3.1 Deselecting Intervals...

The **Interval…** button provides the mechanism to tailor a **Frequency-based** request by overriding selected intervals (**Note**: the initial resource reservation must be saved prior to tailoring frequency-based requests.). Selecting the **Interval…** button, displays a secondary screen that provides a pair of lists: **Unselected Intervals** and **Selected Intervals**. **Unselected Intervals** lists the dates that will not be reserved for the reservation request. **Selected Intervals** lists the dates that will be included for the request. The Selected Interval dates are automatically generated by the system, based upon the **Frequency** option selected (see Section 12.3.2.2, above). You can move them to or from the **Unselected Intervals** list to modify the automated list. Dates are moved from one list to the other by selecting the dates and using the arrow keys. The **Request Name** is also displayed.

1 **Single-click** on your selections and **single-click** on the desired arrow to move dates between the **Selected Intervals** and **Unselected Intervals** lists.

2 **Single-click** on one of the following buttons as appropriate:

- **OK** to save the selections and exit the screen.
- **Cancel** to exit the screen **without** saving changes.

12.3.4 Validate or Reject a Resource Reservation Request

All resource reservation requests must be validated and approved before scheduling. Validation is the process whereby a request is checked for completeness, and its purpose is deemed reasonable. After reviewing a resource reservation request, the Resource Planner may choose to consult with appropriate DAAC staff or assign a staff member ("sponsor") to validate a request. When the request is rejected, the status of the request is changed to "rejected" on the screen.

Table 12.3-6 presents (in a condensed format) the steps required to validate or reject a Resource Reservation Request. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).

- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI, **single-click** on the resource reservation request to be modified.

3 **Single-click** on the **Modify…** button to access the **Resource Reservation Request Edit** GUI.

4 **Single-click** on either **Validated** or **Rejected**.

- **Validated** indicates that the reservation request is complete and 'makes sense'; that is, the request includes the appropriate resources consistent with the type of activity that is being proposed.
- **Rejected** indicates that the reservation request is rejected.
- At this time, the **Comment** field may also be updated.
- The **Status** field contains the status of the reservation request.
  - Status is system-generated based on operator-input in other fields.

5 After data is entered, **single-click** on the appropriate button(s):

- **Save** to save data.
- **Clear** to clear entries. Once cleared, the entries are deleted from the system.
- **Cancel** to exit screen.

Table 12.3-6. Validate or Reject a Resource Reservation Request - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**<resource reservation request>** (to be modified)	**single-click**
3	**Modify…** button	**single-click**
4	**Validated** button or **Rejected** button as appropriate	**single-click**
5	**Save** button	**single-click**

12.3.5 Approve a Resource Reservation Request

The **Approve** button is used when all reviews that are a part of the resource planning process have taken place and there are no objections to the resource usage as described by the request. Clicking on this button will verify that there are no conflicts between this resource reservation and other reservations. If conflicts are detected, a screen will pop up listing the conflicts to be addressed for resolution. Click **OK** to collapse the pop-up screen. Clicking on **Approve** generates the pop-up screen again (if conflicts exist). Approval occurs after a request has been validated and the event time is acceptable.

Table 12.3-7 presents (in a condensed format) the steps required to approve a Resource Reservation Request. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).

- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI, **single-click** on the resource reservation request to be approved.

3 **Single-click** on the **Approve** button.

- If there are resource conflicts resulting from the attempt to approve the resource reservation request, a pop-up dialogue box appears indicating that the approval failed and making reference to the **Message Handler** GUI for further information.

4 **Single-click** on the **OK** button to collapse the pop-up dialogue box.

- If there are no resource conflicts to be resolved, the entry in the Status column of the **Resource Scheduler** GUI indicates that the request is "Approved" (changes from “Validated”). [End of procedure.]
- If there are resource conflicts to be resolved, the entry in the Status column of the **Resource Scheduler** GUI indicates that the request has "Conflicts" (changes from “Validated”). [Continue with Step 5.]

5 If there are resource conflicts to be resolved, examine the information displayed on the **Resource Scheduler** GUI.

- Although the pop-up dialogue box makes reference to the **Message Handler** GUI for further information, no relevant data seems to be displayed there. Therefore, it is more appropriate to check for conflicts in the duration and frequency information for the resource reservation requests displayed on the **Resource Scheduler** GUI. When more than one resource reservation request is scheduled for the same date and time, there may be a conflict (if the same resource is specified in the requests).
- It may be necessary to examine individual resource reservation requests in detail. If so, use the procedure to **Edit a Resource Reservation Request** (Section 12.3.3).

6 If necessary, consult with the resource requester(s), Resource Manager and other personnel to determine which resource reservation request(s) to modify or delete in order to create a conflict-free resource plan.

7 If applicable, go to the procedure to **Delete a Resource Reservation Request** (Section 12.3.8) and delete resource reservation request(s) as necessary to resolve the conflicts.

8 If applicable, go to the procedure to **Edit a Resource Reservation Request** (Section 12.3.3) and modify/validate resource reservation request(s) as necessary to resolve the conflicts.

9 If applicable, return to Step 2 to approve a modified resource reservation request.

- The modified procedure must have been "validated." If necessary, refer to the procedure to **Validate or Reject a Resource Reservation Request** (Section 12.3.4).

Table 12.3-7. Approve a Resource Reservation Request - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**<resource reservation request>** (to be approved)	**single-click**
3	**Approve** button	**single-click**
4	**OK** button	**single-click**
5	If there are resource conflicts to be resolved, examine the information displayed on the **Resource Scheduler** GUI	**read text**
6	Resolve conflicts as necessary	Use procedures in Sections 12.3.8, 12.3.3, and/or 12.3.4

12.3.6 Commit Resource Reservation Requests

Clicking on the **Commit globally** button commits all approved reservation requests and makes them accessible to Production Planning. All committed resource reservations are automatically included in the next production plan to be activated through the Planning Workbench and are subsequently sent to Data Processing. Note that resource reservations/ground events cannot take effect until they have been sent to Data Processing as part of an activated production plan. (Refer to Section 13, Production Planning, to see how production plans are created and activated.)

In Data Processing a "ground event" job for each resource reservation is sent to the specified resource(s) at the indicated start time. If a data processing job is already using the specified resource(s) at the ground event's scheduled start time, the data processing job runs to completion before releasing the resource(s) to the ground event job.

Table 12.3-8 presents (in a condensed format) the steps required to commit a Resource Reservation Request. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 From the **Resource Scheduler** GUI, **single-click** on the **Commit globally** button.

- Status shows **Committed** for all previously **Approved** requests.

2 To view a graphical representation of the resource plan execute the procedure in Section 12.3.7.

3 To exit from the **Resource Scheduler** GUI execute the procedure in Section 12.3.9.

NOTE: Resource reservations/ground events are not sent to data processing and cannot be implemented until they have been included in a production plan. Refer to Section 13, Production Planning, for the procedure on creating (including activating) production plans.

Table 12.3-8. Commit Resource Reservation Requests - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**Commit globally** button	**single-click**
2	View a graphical representation of the resource plan if desired	Use the procedure in Section 12.3.7
3	Exit from the **Resource Scheduler** GUI if desired	Use the procedure in Section 12.3.9

12.3.7 Review the Resource Timeline

The Resource Planning utilities allow the operator to view the Resource Plan as a timeline. Table 12.3-9 presents (in a condensed format) the steps required to review the Resource Timeline. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).

- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI **single-click** on the **Timeline** button.

- The **Resource Timeline** GUI is displayed.

- The display represents a set of resources, arranged along the left side of the screen and some period of time as indicated across the top edge of the screen.
- The use of a resource over a period of time is represented by one or more ‘resource reservation’ bars across the screen.
- A bar represents a time period during which a resource reservation has been planned for the resource.
  - Each bar has the name of the resource reservation and a brief description.
  - For time periods during which a reservation has not been placed against a resource, that resource is planned for use by a default activity, e.g., science processing computers will be used for science processing unless a reservation has been placed against that resource .
  - Scroll bars allow scrolling up and down through the full list of resources and left and right in time.

3 Adjust the **Resource Timeline** window size and the view of the timeline as necessary using the mouse.

- Grab a corner of the timeline window with the cursor and resize the window as desired.
- Scroll up or down through the full list of resources.
- Scroll left or right to go backward or forward in time.

4 If a different time scale (start and end dates and times) is desired, perform Steps 5 through 7; otherwise, go to Step 8.

5 Execute the following menu path:

Display → Change Time Scale

- The **plan window edit** window is displayed.

6 In the **Plan Win Start** and **Plan Win End** fields of the **plan window edit** window enter the date and time for the desired start and end times using the following format:

<DD MMM YYYY hh:mm:ss>

7 When the appropriate date and time have been entered, **single-click** on the appropriate button from the following selections:

- **OK** - to accept the changes and dismiss the **plan window edit** window.
- **Apply** - to accept the changes without dismissing the **plan window edit** window.
- **Cancel** - to cancel the changes and dismiss the **plan window edit** window.

8 If a different time span is desired, **single-click** and **hold** on the **Show** option button, **move** the mouse cursor to the desired selection (highlighting it), then **release** the mouse button.

- Options are: **1 hr, 4 hr, 8 hr, 12 hr, 24 hr, 48 hr, 4 day, 1 week, 2 week, 1 month, full scale.**

9 If no resources are displayed on the GUI or if different resources should be displayed, perform Steps 10 through 14; otherwise, go to Step 15.

10 Execute the following menu path:

Display → Change resources

- The **Resource edit** window is displayed.

11 If adding resource(s) from the **Available Resources** list to the **Viewed Resources** list, select (highlight) the resource(s) to be added, then click on the **Add** button to move the resource(s) to the **Viewed Resources** list.

- Highlighted resource(s) appear(s) on the **Viewed Resources** list.

12 If deleting resource(s) from the **Viewed Resources** list, select (highlight) the resource(s) to be removed, then click on the **Del** button to remove the resource(s) from the **Viewed Resources** list.

- Highlighted resource(s) disappear(s) from the **Viewed Resources** list.

13 If changing the order in which resources are listed in the **Viewed Resources** list, select (highlight) the resource to be moved, then **single-click** on the up or down arrow as necessary to reposition the selected resource.

- Highlighted resource changes position in the **Viewed Resources** list.

14 When the **Viewed Resources** list contains the desired set of resources, **single-click** on the appropriate button from the following selections:

- **OK** - to accept the changes and dismiss the **Resource edit** window.
- **Apply** - to accept the changes without dismissing the **Resource edit** window.
- **Cancel** - to cancel the changes and dismiss the **Resource edit** window.

15 If different color coding of the timeline is desired, perform Steps 16 through 20; otherwise, go to Step 21.

16 Execute the following menu path:

Display → Change colors

- The **Color Selections** window is displayed.

17 **Single-click** on the name of one of the resource reservations to be recolored.

- The resource reservation is highlighted.

18 **Single-click** on the desired color (in the color palette) to be applied to the highlighted resource reservation.

19 Repeat Steps 17 and 18 as necessary.

20 When the appropriate color changes have been made, **single-click** on the appropriate button from the following selections:
- **OK** - to accept the changes and dismiss the **Color Selections** window.
- **Apply** - to accept the changes without dismissing the **Color Selections** window.
- **Cancel** - to cancel the changes and dismiss the **Color Selections** window.

21 Observe the resource reservation information displayed on the **Resource Timeline** GUI.

22 Repeat the previous steps as necessary.

23 If it becomes necessary to exit from the timeline GUI execute the following menu path:
File → Quit

Table 12.3-9. Review the Resource Timeline - Quick-Step Procedures (1 of 2)

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**Timeline** button	**single-click**
3	**Display → Change Time Scale**	**single-click**
4	**<plan window start date and time>**	**enter text**
5	**<plan window end date and time>**	**enter text**
6	**Ok** button	**single-click**
7	**<time span>**	**single-click**
8	**Display → Change Resources**	**single-click**
9	**<resources>** (to be viewed)	**single-click**
10	**Add** button	**single-click**
11	**<viewed resource>** (to be reordered)	**single-click**
12	**<up>** arrow or **<down>** arrow (as necessary to reorder viewed resources)	**single-click**
13	**Ok** button	**single-click**
14	**Display → Change Colors**	**single-click**
15	**<resource reservation>** (to be recolored)	**single-click**
16	**<color>** for resource reservation	**single-click**
17	**Ok** button	**single-click**

Table 12.3-9. Review the Resource Timeline - Quick-Step Procedures (2 of 2)

Step	What to Enter or Select	Action to Take
18	Observe the resource reservation information	**read text**
19	**File → Quit** (to quit the timeline)	**single-click**

12.3.8 Delete a Resource Reservation Request

Table 12.3-10 presents (in a condensed format) the steps required to delete a resource reservation request. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 If necessary, launch the **Resource Scheduler** GUI (refer to Section 12.3.1).

- The **Resource Scheduler** GUI is displayed.

2 From the **Resource Scheduler** GUI, highlight (click on) the resource reservation request you want to delete.

3 Execute the following menu path:

File → Delete

- Status shows "Deleted" for the selected request. The resource reservation request is not removed from the database at this point and is available for future reporting but will have no impact on resource planning. Resource reservations are removed from the Resource reservations (PDPS) database through routine database maintenance activities.

4 To exit from the **Resource Scheduler** GUI execute the procedure in Section 12.3.9.

Table 12.3-10. Delete a Resource Reservation Request - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Launch the **Resource Scheduler** GUI (if necessary)	Use procedure in Section 12.3.1
2	**<resource reservation request>** (to be deleted)	**single-click**
3	**File → Delete**	**single-click**
4	Exit from the **Resource Scheduler** GUI if desired	Use procedure in Section 12.3.9

12.3.9 Shut Down the Resource Scheduler

When resource scheduling activities have been completed, the Message Handler, System Name Server, and Resource Model should be shut down to eliminate unneeded processes and allow other operators to gain access to the resource planning applications. If any of the three processes remains active, it is likely to interfere with subsequent attempts to launch resource planning applications.

Shutting down the Resource Scheduler starts with the assumption that the **Resource Scheduler** GUI has been launched and is currently being displayed.

Table 12.3-11 presents (in a condensed format) the steps required to shut down resource scheduling applications. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 To exit from the **Resource Scheduler** GUI when resource planning activities have been completed execute the following menu path:

File → Exit

- The **Resource Scheduler** GUI disappears unless there are resource reservation requests with a status of "approved".
- If there are any resource reservation requests with a status of "approved" listed on the **Resource Scheduler** GUI, a **Close Application** pop-up dialogue box is displayed with a message "Status of the listed reservations" and a list of the resource reservation requests with "approved" status.

2 If the **Close Application** pop-up dialogue box is displayed, **single-click** on the appropriate button from the following selections:

- **Ok** - to quit the **Resource Scheduler** GUI and dismiss the dialogue box.
  - Selecting **Ok** effectively commits all "approved" Resource Reservations.
- **Cancel** - to dismiss the dialogue box and return to the **Resource Scheduler** GUI.

3 After quitting the **Resource Scheduler** GUI **single-click** in the UNIX window used to start the resource scheduling applications.

4 Shut down the Message Handler, System Name Server, and Resource Model by entering:

EcPlRpSlayAll <MODE> <application_id>

- The **Message Handler** GUI disappears.

5 To obtain a list of active processes in the specified mode enter:

ps -ef | grep <MODE>

- A list of active processes in the specified mode is displayed.
- If an error message is received when **ps -ef | grep <MODE>** is entered, enter:

  ps -auxwww | grep <MODE>

6 Examine the list of processes running in the specified mode to determine whether the Message Handler, System Name Server, and Resource Model processes have actually been shut down.

- None of the following processes should be active:
  - EcPlRpRe
  - EcPlRpSi
  - EcPlRpTl
  - EcPlMsh
  - EcPlSns
  - EcPlRpRm

7 If any of the specified processes [especially the Message Handler, System Name Server, and/or Resource Model process(es)] is/are still active, terminate the active process(es)by entering:

kill -15 <process ID1> [<process ID2>] [<process ID3>] […]

8 Repeat Steps 5 through 7 as necessary.

Table 12.3-11. Shut Down the Resource Scheduler - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**File** → **Exit** (to quit the **Resource Scheduler** GUI)	**single-click**
2	**OK** button	**single-click**
3	UNIX window (Planning/Management Workstation)	**single-click**
4	**EcPlRpSlayAll <MODE>** <application_id>	**enter text, press Enter**
5	**ps -ef | grep <MODE>**	**enter text, press Enter**
6	Identify the resource scheduling process(es) that has (have) not shut down	**read text**
7	**kill -15 <process ID1> [<process ID2>] [<process ID3>] […]** if necessary	**enter text, press Enter**

12.4 Tuning System Parameters

The values assigned to system parameters affect the functioning and performance of the system. When certain parameters are modified, the system operates differently. Changes to some other parameters may not appear to affect the system although there may in fact be subtle effects. In any case before system parameters are modified it is essential to understand what will happen to system functioning and performance.

Many system parameters may be subject to control by Configuration Management (CM). When making or requesting a change to system parameters, the CM process at the particular site must be followed (if applicable).

Values are assigned to Data Processing Subsystem and Planning Subsystem parameters in the following databases:

- PDPS database.
- Configuration Registry database.

The Configuration Registry Server provides a single interface (via a Sybase server) for retrieving configuration attribute-value pairs for ECS servers from the Configuration Registry database. When ECS servers are started, they access the Configuration Registry Database to obtain needed configuration parameters.

The Database Administrator has access to a Configuration Registry GUI for viewing and editing configuration data in the database. Therefore, it is necessary to coordinate with the Database Administrator when changes to configuration parameters are needed. Also, as previously mentioned, changes to configuration-controlled parameters are subject to approval through the site CM process.

Default and adjusted values assigned to system parameters vary from site to site. For guidance concerning the assignment of values to parameters included in the Configuration Registry refer to document 910-TDA-022, *Custom Code Configuration Parameters for ECS*. The document is available at http://cmdm.east.hitc.com/baseline/ under “Technical Documents.”

The following parameters are examples of parameters whose values may be modified to enhance system functioning or performance:

- AppLogSize [parameter applies to all servers].
  - Maximum size of the application log (ALOG) file for a particular application.
  - Recommended size varies considerably depending the nature of the application for which the file is being written.
- AppLogLevel [parameter applies to all servers].
  - Level of detail provided in the ALOG file for a particular application.
  - Acceptable values are 0, 1, 2, or 3.
  - A setting of “0” provides the most data.
- DebugLevel [parameter applies to all servers].
  - Level of detail provided in the debug log file for a particular application.
  - Normally acceptable values are 0, 1, 2, or 3.
  - A setting of "0" turns off logging; a setting of “3” provides a significant amount of data.
- DpPr_MAX_RETRIES [EcDpPrEM and EcDpPrDeletion parameter (also EcDpPrQaMonitorGUI and several Science Software Integration and Test programs)].
  - Number of retries (e.g., 30) to the Science Data Server for acquires/inserts before giving up.

- DpPr_WAIT_PERIOD [EcDpPrEM and EcDpPrDeletion parameter (also EcDpPrQaMonitorGUI and several Science Software Integration and Test programs)].
  - Time in seconds (e.g., 120) to wait between retries to the Science Data Server.
- DpPrRM_MAX_RETRIES [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter].
  - Maximum number (e.g., 100) of attempts to allocate a computer resource.
- DpPrRM_RETRY_PERIOD [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter].
  - Number of seconds (e.g., 120) between retries when trying to allocate a resource.
- DpPrMaxConcurrentDPRs [EcDpPrJobMgmt parameter].
  - Maximum allowed jobs.
  - Three integer values (e.g., 100 100 100) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.
- DpPrMinConcurrentDPRs [EcDpPrJobMgmt parameter].
  - Minimum allowed jobs.
  - Three integer values (e.g., 0 0 0) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.
  - Minimum number of concurrent DPRs for each job class (i.e., routine, on demand, reprocessing) NOT CURRENTLY USED.
- DpPrAutoSysMaxDPRs [EcDpPrJobMgmt parameter].
  - Total number of jobs (e.g., 100) allowed in AutoSys.
- DpPrDeleteFailedPGEJobs [EcDpPrJobMgmt parameter].
  - If TRUE, failed PGE Jobs are removed by Job Management, as necessary, when space is needed for another job that is ready to run. This is recommended to keep job management straightforward. However, this may be confusing for the operator, since they may not get a chance to see the failure if the system is busy.
  - If FALSE (the usual value), failed PGE Jobs are left in AutoSys. They must not be removed manually from AutoSys, however, since they will be removed by the Production Request Editor when a Production Request or DPR is cancelled.
- DBConnections [EcPoConnections (includes EcPlSubMgr, EcPlOdMgr, EcDpPrDeletion, EcDpPrJobMgmt and EcDpPrJobMgmtClient) parameter].
  - Number of connections needed by a particular application (e.g., 10 for EcPlOdMgr).
  - Optional parameter that specifies the number of connections to maintain in the connection pool.
  - The parameter is a list of positive integers. There must be one entry for each DbHandle in the DbHandleList.
  - Generally it should be set to the maximum number of connections that are expected to be used simultaneously in a process. If one connection per thread is used, this will be the same as the number of concurrent threads expected to

execute. When the pool is used up there is a performance penalty to allocate and deallocate connections on the fly.
  - If this parameter is not specified or is given as “NONE”, it defaults to 1.
- SleepDelayForFailures [EcPlSubMgr parameter].
  - Amount of time in seconds (e.g., 60) to wait before reprocessing failed notifications. If the specified value is less than 60, a default value of 60 seconds would be assumed.
  - Duration of the sleep delay used by the failed notification thread in seconds.
  - Less frequent checking can increase speed for the other threads.
- SleepDelayForTimers [EcPlSubMgr parameter].
  - Amount of time in seconds (e.g., 60) the Subscription Manager should sleep between checking for expired timers. It should be set to the minimum amount of time a timer will be set for at this DAAC. The minimum it can be set to is 60 seconds.
  - Duration of sleep delay used by the timer checking thread in seconds.
  - Less frequent checking can increase speed for the other threads.
- SleepDelayForExp [EcPlOdMgr parameter].
  - Sleep delay for expiration thread in seconds (e.g., 86400).
  - Should be considerably greater than the sleep delay for completion threads (SleepDelayForCmp).
- SleepDelayForCmp [EcPlOdMgr parameter].
  - Sleep delay for completion threads in seconds (e.g., 300).
  - Should be considerably less than the sleep delay for expiration threads (SleepDelayForExp).
- SocketLimit [EcDpPrDeletion, EcDpPrJobMgmt, EcPlOdMgr, EcPlSubMgr parameter].
  - Number of connections (e.g., 200) to a server through the Hubble Space Telescope (HST) sockets middleware.
  - Too low a number misses connections.
  - Too high a number may adversely affect the memory of the server's host.

NOTE: When the value assigned to a parameter has been changed and saved in the Configuration Registry, the modified value does not take effect until the affected server has been restarted. For example, if the debug level for the Subscription Manager log has been changed from “2” to “3” in the Configuration Registry, the modification does not affect the recording of data in the log until after a warm restart of the Subscription Manager (at which time the server would read the parameters in the Configuration Registry).

Table 12.4-1, below, provides an Activity Checklist table of System Tuning activities.

Table 12.4-1. Tuning System Parameters - Activity Checklist

Order	Role	Task	Section	Complete?
1	Resource Planner/ Production Planner/ Production Monitor	Monitor the Load on Processing Resources	(P) 12.4.1	

12.4.1 Monitor the Load on Processing Resources

The Production Planner and Production Monitor should work with the Resource Planner to make optimum use of processing resources. The Resource Planner allocates the disk partitions, CPUs, and RAM available for processing among the active modes (e.g., OPS, TS1, TS2). The Production Planner and Production Monitor monitor the load on the processing resources.

The Resource Planner assigns the bulk (typically 60% - 80%) of the processing resources to the OPS mode. The remainder of the processing assets are divided among the modes used for SSI&T and new version software checkout.

The Production Planner and Production Monitor monitor the load on the processing resources to identify whether the actual load is appropriately distributed among modes. They inform the Resource Planner of under- or over-use of resources as allocated.

When monitoring the load on the processing resources, the Production Planner and Production Monitor should take the following considerations into account:

- Disk space allocated to OPS mode is likely to be used to capacity.
- Disk space assigned to the other two modes may not fill up.
- There is no one-to-one mapping of CPU allocation with actual CPUs on the science processor.
- The operating system (OS) takes care of true CPU and RAM allocation.
  - Actual CPU usage during processing is limited by the OS.
  - If ten CPUs have been specified for a particular mode, only ten Data Processing Requests (DPRs) can be running the Execute job at a given time.
  - What is really being defined is the maximum number of DPRs that will execute at a given time.
- CPUs can be over-allocated or under-allocated as necessary to get the most out of the CPUs on each science processor.
- If monitoring indicates that the processor is underused when OPS mode is at full processing capacity, the number of CPUs allocated to OPS mode could probably be increased.
- If the science processor is at full capacity when OPS mode is at full processing capacity (and the processor may be overworked) the number of CPUs allocated to OPS mode should be reduced.
- Random-access memory (RAM) is subject to the same considerations as CPUs.

– RAM can be over-allocated or under-allocated as necessary to get the most out of the memory on each science processor.

12.4.2 Strategies for Tuning

A scenario that demonstrates how DPRs might be processed under a particular set of conditions and some strategies for tuning the system are presented in the paragraphs that follow. The processing conditions include the following types of items:

- The total number of jobs allowed into AutoSys.
- The number of CPUs available for processing.
- Characteristics of the PGEs to be processed.

The total number of jobs allowed into AutoSys is controlled by the DpPrPgeLimits table in the PDPS database. An example of some of the types of data maintained in the DpPrPgeLimits table is shown in Table 12.4-2.

Table 12.4-2. Example of PDPS Database DpPrPgeLimits Table Contents (Selected Columns)

computerName [Virtual Computer]	pgeId	maxConcurrent [DPRs]
A	1	20
B	1	20
A	2	20
B	2	20

The scenario assumes that each of the virtual computers (i.e., A and B) listed in Table 12.4-2 has 16 CPUs. (There are 32 CPUs total.)

Relevant PGE characteristics are shown in Table 12.4-3.

Table 12.4-3. PGE Characteristics

PGE	# CPUs Used	Average Execution Time	Average Stage Time	Destage Time
1	1	5 minutes	5 minutes	5 minutes
2	1	60 minutes	5 minutes	5 minutes

Assuming that 100 DPRs of each type (i.e., PGE 1 and PGE 2 - 200 DPRs total) are ready to run and are released at once into AutoSys, the following actions occur:

- Eighty (80) DPRs enter AutoSys. The remaining 120 are queued, with their assignments already made:

  - Machine (Virtual Computer) A: 20 PGE 1s start staging; 30 PGE 1s are queued on Machine A; 20 PGE 2s start staging; 30 PGE 2s are queued on Machine A.
  - Machine (Virtual Computer) B: 20 PGE 1s start staging; 30 PGE 1s are queued on Machine B; 20 PGE 2s start staging; 30 PGE 2s are queued on Machine B.
- After about five (5) minutes, all 80 DPRs that were staging have finished staging and are ready for execution. However, only 32 CPUs are available.
- The first 32 DPRs that ask for CPUs get them and start running [sixteen (16) on Machine A and sixteen (16) on Machine B]. Forty-eight (48) DPRs are waiting.
  - Assuming that in the Registry database DpPrRM_RETRY_PERIOD is set to 120 seconds and DpPrRM_MAX_RETRIES is set to 100, the waiting DPRs keep trying every two minutes for up to 100 times each before timing out (after 200 minutes).
  - Note that in this example timing out is a real possibility.
- The quick jobs complete processing after five (5) minutes, freeing up sixteen (16) CPUs. In the current example, the sixteen (16) CPUs are subsequently occupied with about eight (8) five-minute PGEs and eight (8) 60-minute PGEs because CPUs are given randomly to whichever DPR gets back first to asking for them after waiting for the retry period (i.e., 120 seconds). Priorities are not used.
  - At first, there was a 50:50 ratio of fast:slow DPRs, now there is a 25:75 ratio of fast:slow. After another five (5) minutes, the ratio becomes 12.5:87.5 fast:slow, so 87.5 % of the CPUs are occupied by 60-minute DPRs.
- Apparently, the 60-minute DPRs tend to dominate the CPUs. After one (1) hour the first batch of sixteen (16) 60-minute PGEs vacates the CPUs to be replaced by eight (8) five-minute PGEs and eight (8) 60-minute PGEs, but the five-minute PGEs become extinguished again by the slow ones.
  - If the staging and destaging times were not the same (so the DPRs didn't have the same opportunity to hit the execution stage at the same time) the scenario would proceed differently.

Various strategies can be employed to tune the system:

- Limit the number of DPRs through the use of the DpPrPgeLimitsTable.
  - In the preceding example if the number of slow DPRs allowed into AutoSys is less than the number of CPUs, there is always a channel for the fast jobs to squeeze through.
  - The big disadvantage to this approach is that the slow jobs are also being prevented from staging.
- Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs).
  - This approach allows more of each type of PGE into the science processors.
  - The disadvantage is that it could overwhelm the science computers. However, they are kept busy.
- Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers.
  - This approach is another way to guarantee bandwidth (CPUs) to PGEs.

– The disadvantage of this approach is that some CPUs could remain idle, not being seen by one of the virtual computers.
– In the past, there may have also been some code problems with supporting this, but those difficulties should have been resolved.

Probably some combination of the first two of the preceding strategies is best; i.e., increase the number of declared CPUs to be more than the total number of slow jobs allowed into AutoSys, always leaving some CPUs for a channel of fast jobs. The total number of faster-moving jobs should be increased to make sure that there is always be a queue of them available to get their channel occupied.

The staging and destaging times have to be accounted for and this could change things in terms of using the DpPrPgeLimits table and the number of CPUs per processor to tune the job flow.

Also, it is important to perform regular garbage collection on all of the virtual computers. Procedures for cleaning the PDPS database and DPS disks (i.e., "garbage collection") are provided in Chapter 13, Production Planning.

12.5 Troubleshooting Resource Planning Problems

Troubleshooting is a process of identifying the source of problems on the basis of observed trouble symptoms. One common source of problems involves the reliance on messages or data from other subsystems. However, unlike many other operational areas in ECS, Resource Planning does not have interfaces with many other subsystems. Consequently, problems with Resource Planning can usually be traced to either some part of the Planning Subsystem or the ECS infrastructure.

Table 12.5-1, below, provides an Activity Checklist for troubleshooting Resource Planning problems.

Table 12.5-1. Troubleshooting Resource Planning Problems - Activity Checklist

Order	Role	Task	Section	Complete?
1	Resource Planner	Troubleshoot a Resource Planning Problem	(P) 12.5.1	
2	Resource Planner	Check Log Files	(P) 12.5.2	
3	Resource Planner	Check Database Connections	(P) 12.5.3	

Fault Recovery

Each request that crosses a client/server boundary is assigned a system-unique identifier referred to as an RPC ID. (RPC refers to Remote Procedure Call, the mechanism by which requests are submitted from client to server.) The RPC ID facilitates the automatic fault recovery events that occur whenever there is a client or server failure.

- As a request propagates through the system, each associated client/server exchange is assigned a unique RPC ID.
  - The RPC ID for each interaction is derived from the previous RPC ID received by the client for the request. Consequently, all RPC IDs associated with a given request have a common portion that relates the various client/server calls to one another.
  - Given the previous RPC ID, clients consistently reproduce the same RPC ID that was submitted to the server on the subsequent event.
- The concept of reproducible RPC IDs is central to the ECS fault recovery capability.
  - When requests are retried from client to server, they are always submitted with the same RPC ID that was used in the original submission of the request, even if either the client or server has crashed between retries.
- The RPC ID is also central to the check-pointing aspect of fault recovery.
  - As requests arrive at fault recovery-enabled servers, they are recorded in a persistent store (typically a database), tagged with the RPC ID, which identifies the request.
  - As the request is serviced, check-pointing state information may be updated in the persistent store, up to and including the completion status of the request.
  - This allows the servers to resume servicing from the last check-pointed state, particularly upon resubmission from a client.

Fault Handling

Failure events are classified according to the following three severity levels:

- Fatal error.
  - Returned when a request cannot be serviced, even with operator intervention.
  - For example, if a request is made to distribute data via ftp to a non-existent host, the request is failed with a fatal error.
- Retry error.
  - Potentially recoverable error.
  - Normally, a retry error would be returned to the client only when the server cannot recover from the error automatically.
  - A retry error may require operator assistance during recovery. For example, a tape left in a tape drive might have to be removed manually.
- Warning.
  - Provided when operations can proceed without interruption, but an unexpected circumstance was detected.
    - For example, when using the Resource Scheduler GUI, the Resource Planner would enter a new name for a resource reservation request after being notified that there was a previously existing resource reservation request with the name that had been entered.

Transient errors (such as network errors) are always retry errors.

- In general, clients and servers that experience transient retry errors first attempt to recover by retrying the operation automatically.
- One special case of this is "rebinding," which refers to the process by which a client automatically attempts to re-establish communication with a server in the event communication is disrupted.
  - The disruption may be caused by transient network failure, or by the server crashing or being brought down.
  - In any case, the client automatically attempts to reconnect to the server for a configurable period of time on a client-by-client basis.

ECS processes encountering an error or receiving an error from a server request can either pass the error back to a higher-level client or present it to the operator for operator intervention.

Client Crash and Restart

In general when a client crashes, the server continues to service the requests that were in process at the time of the client's crash. When a client restarts in the ECS system, it sends a restart notification to each server with which it interacts.

- Clients notify servers that they have come up either "cold" or "warm."
- Generally, the notification temperature sent to the server matches the temperature at which the client process is restarted.

The default server behavior in response to startup notification from a client is as follows:

- Warm Notification.
  - Outstanding requests for the restarted clients remain available in the persistent store.
  - The outstanding requests may be resubmitted by the client, and are serviced to completion upon resubmission.
  - Associated resources are left allocated until the requests are completed.
- Cold Notification.
  - All outstanding requests for the restarted client are cancelled.
  - If the client resubmits any cancelled request using the same RPC ID (e.g., by pressing the Retry button from an operator GUI), it is failed with a fatal error due to the client cold startup notification.
  - Any resources associated with the cancelled requests are released and reclaimed by the system.

Server Crash and Restart

When a server crashes, clients cannot continue to submit requests for processing.

- Synchronous requests in progress result in a Distributed Computing Environment (DCE) exception being thrown back to the client process, which enters a rebinding failure recovery mode (as previously mentioned).

- Attempts to submit requests while the server is down result in the client blocking until a communication timeout has been reached.
- Although DCE has been replaced by socket-based library calls (i.e., CCS Middleware), the DCE exception code is handled by the CCS Middleware.

When a server restarts, it may perform various resynchronization activities in order to recover from an unexpected termination.

- In the event of a server cold start or cold restart, the server typically cancels all outstanding requests and reclaims all associated resources.
- In general, existing request queues are retained for warm restarts and cleared for cold starts or cold restarts.

12.5.1 Troubleshoot a Resource Planning Problem

1 If it is not possible to log in to the Planning Subsystem host, ask the Operations Controller/System Administrator to verify that the host is “up."

- Examples of Planning/Management Workstation host names include **e0pls03**, **g0pls01,** and **l0pls02.**

2 If the GUI (i.e., the **Resource Editor** GUI or the **Resource Scheduler**) is not displayed when the start-up script has been properly invoked, ensure that the DISPLAY variable was set properly.

- For detailed instructions refer to the applicable procedure.
  - **Launch the Resource Editor** (Section 12.2.2).
  - **Launch the Resource Scheduler** (Section 12.3.1).

3 If an error message is received indicating that SNS (System Name Server) and/or Resource Model is/are in use using the selected Application ID and if working in a different mode from the person using the selected Application ID, use a different Application ID.

- For detailed instructions refer to the applicable procedure.
  - **Launch the Resource Editor** (Section 12.2.2).
  - **Launch the Resource Scheduler** (Section 12.3.1).

4 If an error message is received indicating that SNS (System Name Server) and/or Resource Model is/are in use using the selected Application ID and if working in the same mode as the person using the selected Application ID, coordinate use of Planning applications with the other user and/or the System Administrator.

5 If an error message associated with the Resource Editor is received, refer to Table 12.5-2, Resource Editor User Messages.

- The table is adapted from the corresponding table in 609-CD-610-003, *Release 6B Operations Tools Manual for the ECS Project*).

6 If an error message associated with the Resource Scheduler is received, refer to Table 12.5-3, Resource Scheduler User Messages.

- The table is adapted from the corresponding table in 609-CD-610-003, *Release 6B Operations Tools Manual for the ECS Project*).

7 If some other type of problem is encountered, check the appropriate log file.

- For detailed instructions refer to the **Check Log Files** procedure (Section 12.5.2).

8 If the problem cannot be identified and fixed without help within a reasonable period of time, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 12.5-2. Resource Editor User Messages (1 of 3)

Message Text	Impact	Cause and Corrective Action
A resource with this name already exists - re-enter name	Each resource name in the database must be unique.	1. Enter a different name in the **Resource Name** field. 2. **Single-click** on the **Save** button.
Activity Type is Not initialized	Without this field initialized, the "Save" operation gets rejected.	1. Shut down all Resource Planning tasks. [For detailed instructions refer to the **Shut Down Resource Definition Applications** procedure (Section 12.2.7).] 2. Notify the Database Administrator to have the PDPS database initialized (run the EcPlDbBuild script in the /usr/ecs/<MODE>/CUSTOM/utilities directory). 3. Relaunch Resource Planning applications. [For detailed instructions refer to the **Launch the Resource Editor** procedure (Section 12.2.2).] 4. Resume the operation that elicited the error message.
Block Size must be an integer number – reenter	Integer only.	1. Enter the appropriate integer (e.g., 1024) in the **Block Size** field. 2. **Single-click** on the **Save** button.
Block Size required	This is a required field.	1. Enter the appropriate integer (e.g., 1024) in the **Block Size** field. 2. **Single-click** on the **Save** button.
Error modifying computer resource	Database interface error.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Error modifying computer resource comments	Database interface error.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 12.5-2. Resource Editor User Messages (2 of 3)

Message Text	Impact	Cause and Corrective Action
Error saving computer resource	The operation failed due to an error in the database interface.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Error saving computer resource comments	Database interface error.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Number of cpus must be an integer number	Non-numeric data are not valid.	1. Enter the appropriate integer (e.g., 22) in the **Number of CPUs** field. 2. **Single-click** on the **Save** button.
Number of cpus required	This is a required field.	1. Enter the appropriate integer (e.g., 22) in the **Number of CPUs** field. 2. **Single-click** on the **Save** button.
Operating system required	This is a required field.	1. Enter the appropriate operating system data (e.g., IRIX 6.5.17) in the **Operating System** field. 2. **Single-click** on the **Save** button.
Partition Size must be a number - reenter	Integer only.	1. Enter the appropriate integer (e.g., 400000) in the **Partition Size** field. 2. **Single-click** on the **Save** button.
Partition Size required	This is a required field.	1. Enter the appropriate integer (e.g., 400000) in the **Partition Size** field. 2. **Single-click** on the **Save** button.
Resource is reserved - cannot modify	The Resource Scheduler GUI reserves the resource.	If possible, leave the resource alone. However, if the resource definition needs to be modified immediately, use the **Resource Scheduler** to change the status or delete the reservation. 1. If the resource definition needs to be modified immediately, first delete all resource reservations that specify the resource. [For detailed instructions refer to the **Delete a Resource Reservation Request** procedure (Section 12.3.8).] 2. Modify the resource definition. [For detailed instructions refer to the **Modify a Resource** procedure (Section 12.2.5).]
Resource name required	Each resource in the database must have a unique name.	1. Enter an appropriate name in the **Resource Name** field. 2. **Single-click** on the **Save** button.
Resources loaded	The resources list has been loaded from the MSS baseline configuration.	For information only. No action is necessary.

Table 12.5-2. Resource Editor User Messages (3 of 3)

Message Text	Impact	Cause and Corrective Action
Resources not loaded - file not found	The MSS baseline configuration file is not found in the previously designated directory.	No Longer Applicable.
Select a resource to modify from the list	The selected resource should be one of the defined resources.	Select (highlight) the resource to be modified in the **Resource Name** list displayed on the **Resource Editor**.
Strings should be selected	AutoSys definition requires the association of a string name.	1. Move string resources between the **Strings** and **Associated Strings** lists as necessary by selecting (highlighting) the string to be moved, then **single-clicking** on the right or left arrow button (as applicable) to move the string to the other list. 2. **Single-click** on the **Save** button.
Total ram must be an integer number	Integer only.	1. Enter the appropriate integer (e.g., 1000) representing the computer's total RAM (in megabytes) in the **Total RAM** field. 2. **Single-click** on the **Save** button.
Total ram required	This is a required field.	1. Enter the appropriate integer (e.g., 1000) representing the computer's total RAM (in megabytes) in the **Total RAM** field. 2. **Single-click** on the **Save** button.
Unable to lock Resource tables - cannot delete resource	The processing software uses the resource or its member resource.	Do not delete the resource definition at this time. The resource definition cannot be deleted while the resource is in use. 1. Wait until the resource has been released by Data Processing. 2. Try again to delete the resource definition. [For detailed instructions refer to the **Delete a Resource** procedure (Section 12.2.6).]
Unable to lock Resource tables - cannot modify resource	The processing software uses the resource or its member resource.	Do not modify the resource definition at this time. The resource definition cannot be modified while the resource is in use. 1. Wait until the resource has been released by Data Processing. 2. Try again to modify the resource definition. [For detailed instructions refer to the **Modify a Resource** procedure (Section 12.2.5).]

Table 12.5-3. Resource Scheduler User Messages (1 of 4)

Message Text	Impact	Cause and Corrective Action
A Reservation must be selected to delete	Operator cannot proceed.	1. If the desired resource reservation request is not included in the list displayed on the **Resource Scheduler**, **single-click** and **hold** on the **Activity Type** option button and select the appropriate category of activity (or select **All**) from the option menu that is displayed. 2. **Single-click** on the resource reservation request to be deleted. 3. Execute **File** → **Delete** from the **Resource Scheduler** pull-down menu.
A Reservation must be selected to modify	Operator cannot proceed.	1. If the desired resource reservation request is not included in the list displayed on the **Resource Scheduler**, **single-click** and **hold** on the **Activity Type** option button and select the appropriate category of activity (or select **All**) from the option menu that is displayed. 2. From the **Resource Scheduler**, **single-click** on the resource reservation request to be modified. 3. **Single-click** on the **Modify…** button to access the **Resource Reservation Request Edit/Definition** GUI.
Can't insert new ResvName: <name> into database	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
can't send requestActChg to resource model for resvName: <name>	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Delete ResvName: <name> from the list	The database cannot be updated.	Delete the resource reservation request. [For detailed instructions refer to the **Delete a Resource Reservation Request** procedure (Section 12.3.8).]
Error in creating a new object for row: <row>.	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 12.5-3. Resource Scheduler User Messages (2 of 4)

Message Text	Impact	Cause and Corrective Action
Fail to modify resvName: <name>	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
I can't find <Plan Name>.	There is a problem with the resource pool.	Enter a valid Plan Name.
Must select reservation	Operator cannot proceed.	1. If the desired resource reservation request is not included in the list displayed on the **Resource Scheduler**, **single-click** and **hold** on the **Activity Type** option button and select the appropriate category of activity (or select **All**) from the option menu that is displayed. 2. From the **Resource Scheduler**, **single-click** on the desired resource reservation request.
New Resvation can't leave resources list of ResvName: <name> empty	This required field must be filled.	Ensure that there is at least one entry in the **Selected Resources** list on the **Resources Selection** GUI. [For detailed instructions refer to the **Selecting Resources...** procedure (Section 12.3.2.1).]
Open one Reservation at a time, Please	Reservation cannot be opened.	1. Find and **single-click** on the open **Resource Reservation Edit/Definition** GUI that is already open. 2. **Single-click** on either the **Save** or **Cancel** button (as appropriate). 3. Open the desired resource reservation request. [For detailed instructions refer to either the **Create a Resource Reservation Request** procedure (Section 12.3.2) or the **Edit a Resource Reservation Request** procedure (Section 12.3.3).]
PlRpSiScheduler::modifyReservation – can't save new info for resvName: <name>.	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
ResvName: < > already has status <status>.	Action cannot be completed.	Choose a different action.
ResvName: <name> can't replace new Interval List	The database cannot be updated.	1. Type a new (unique) name for the resource request in the **Request Name** field (**Resource Reservation Request Edit/Definition** GUI). 2. Click on the **Save** button. [For detailed instructions refer to r the **Create a Resource Reservation Request** procedure (Section 12.3.2).]

Table 12.5-3. Resource Scheduler User Messages (3 of 4)

Message Text	Impact	Cause and Corrective Action
ResvName: <name> Selected Intervals list can't be empty	This required field must be filled.	Ensure that there is at least one entry in the **Selected Intervals** list on the **Intervals Selection** GUI. [For detailed instructions refer to the **Deselecting Intervals...** procedure (Section 12.3.3.1).]
ResvName: <name> Selected Resources list can't be empty	This required field must be filled.	Ensure that there is at least one entry in the **Selected Resources** list on the **Resources Selection** GUI. [For detailed instructions refer to the **Selecting Resources...** procedure (Section 12.3.2.1).]
ResvName: <name> accepts new resources list	Informational message.	For information only. No action is necessary.
resvName: <name> can't replace new Resource List	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
ResvName: <name> can't uncommitted < > RActAlls	The database cannot be updated.	1. Check the database connections. [For detailed instructions refer to the **Check Database Connections** procedure (Section 12.5.3).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
resvName: <name> fails to approve - status is changed to <status>.	The plan cannot be approved due to conflicts with other reservations.	Resolve the conflict(s). [For detailed instructions refer to the **Approve a Resource Reservation Request** procedure (Section 12.3.5).]
resvName: <name> myTime: <time> resourceName: <name> conflicted Time: <time> conflictResvName: <name>	Informational message.	For information only. No action is necessary.
ResvName: <name> resource list is less now	Informational message.	For information only. No action is necessary.
ResvName: <name> status is changed from approved to committed	Informational message.	For information only. No action is necessary.
ResvName: <name> status is changed to <status>	Informational message.	For information only. No action is necessary.

Table 12.5-3. Resource Scheduler User Messages (4 of 4)

Message Text	Impact	Cause and Corrective Action
Success to approve reservation Name: <name>	Informational message.	For information only. No action is necessary.
Success to update resvName: <name> name	Informational message.	For information only. No action is necessary.
This Name: <name> with status: <status> has been used, Please pick another Name.	The resource reservation request name must be unique.	1. Enter a different name in the **Request Name** field. 2. **Single-click** on the **Save** button.

12.5.2 Check Log Files

Log files can provide indications of the following types of problems:

- Communication problems.
- Database problems.
- Lack of disk space.

Table 12.5-4 presents (in a condensed format) the steps required to check log files. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 Access a terminal window logged in to the appropriate host.

- Examples of Planning/Management Workstation host names include **e0pls03**, **g0pls01,** and **l0pls02.**
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

2 At the command line prompt enter:

cd /usr/ecs/<MODE>/CUSTOM/logs

- **<MODE>** is current mode of operation.
  - TS1 - Science Software Integration and Test (SSI&T)
  - TS2 - New Version Checkout
  - OPS - Normal Operations
- "logs" is the directory containing resource planning log files (e.g., EcPlRpRe.ALOG, EcPlRpReDebug.log, EcPlRpSi.ALOG, or EcPlRpSiDebug.log).

3 At the command line prompt enter:

pg <file name>

- **<file name>** refers to the resource planning log file to be reviewed (e.g., EcPlRpRe.ALOG, EcPlRpReDebug.log, EcPlRpSi.ALOG, or EcPlRpSiDebug.log.
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **more**, **vi**, **view**) can be used to review the log file.

4 Review the log file to identify problems that have occurred.

- To exit from **pg** at the **:** prompt enter:

  q

  – The command line prompt is displayed.

5 Respond to problems as follows:

- Resource Planning-related problems.
  – Perform the appropriate procedure(s) from Table 12.5-1, Troubleshooting Resource Planning Problems.
- Communication problems.
  – Notify the Operations Controller/System Administrator of suspected communication problems.
- Database problems.
  – Verify that relevant database servers are running.
  – Check for lack of (or corruption of) data in the database using either a database browser or interactive structured query language (isql) commands.
  – Notify the Database Administrator of suspected database problems.
- Lack of disk space.
  – Remove unnecessary files.
  – Notify the Operations Controller/System Administrator of recurring disk space problems.

Table 12.5-4. Check Log Files - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	UNIX window	**single-click** or use procedure in Section 12.2.1
2	**cd /usr/ecs/<MODE>/CUSTOM/logs**	**enter text, press Enter**
3	**pg <file name>**	**enter text, press Enter**
4	identify problems indicated in the log file	**read text**
5	Respond to problems as necessary	

12.5.3 Check Database Connections

If applications (including the GUIs) are unable to connect to the database, data cannot be retrieved or (in the case of the GUIs) displayed. Consequently, if a GUI does not display data or if the display does not refresh, checking the database connections is a logical step in trying to isolate the problem.

Table 12.5-5 presents (in a condensed format) the steps required to check database connections. If you are already familiar with the procedures, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedures:

1 Submit a request to the Database Administrator to identify the values for parameters associated with the appropriate application.
- The following parameters should be requested:
  - **DBName.**
  - **DBServer.**
  - **DBMaxConnections.**
- The preceding parameters are associated with the following applications:
  - EcPlRpRe.
  - EcPlRpRm.
  - EcPlRpSi.
  - EcPlRpTl.

2 Access a terminal window logged in to the Queuing Server host.
- Examples of Queuing Server host names include **e0sps04**, **g0sps06**, and **l0sps03.**
- For detailed instructions refer to the **Log in to ECS Hosts** procedure (Section 12.2.1).

3 At the command line prompt enter:

isql –U <user ID> -S <database server>

- **<user ID>** is the database user's identification; e.g., **pdps_role**.
- **<database server>** is the database server; e.g., **g0sps06_srvr**.

4 At the **Password:** prompt enter:

<database password>

- **<database password>** is the password for logging in to the database using the specified **<user ID>**.
- A **1>** prompt is displayed, indicating that a connection has been made with the database.

5 At the **1>** prompt enter:

sp_who

6 At the **2>** prompt enter:

go

- A listing of connections to the database is displayed.
- The listing includes data in the following columns:
  - **spid.**
  - **status.**
  - **loginame.**
  - **hostname.**
  - **blk.**
  - **dbname.**
  - **cmd.**

7 At the **1>** prompt enter:

sp_configure "user connections"

8 At the **2>** prompt enter:

go

- A listing of connections to the database is displayed.
- The listing includes the following types of data:
  - **Parameter Name** (i.e., number of user connections).
  - **Default.**
  - **Memory Used.**
  - **Config Value.**
  - **Run Value.**

9 To exit from **isql** at the **1>** prompt enter:

quit

- The connection with the database is discontinued.

10 Compare the number of actual connections (results of **sp_who**) with the number of connections for which the database has been configured (results of **sp_configure "user connections"**).

11 If the number of actual connections is very close to the number of connections for which the database has been configured, notify the Database Administrator of the fact.

12 If the number of actual connections is **not** very close to the number of connections for which the database has been configured, compare the number of actual connections with the value for DBMaxConnections that the Database Administrator specified (Step 1).

13 If the number of actual connections is very close to the value for DBMaxConnections, notify the Database Administrator of the fact.

- It may be advisable to increase the value assigned to the DBMaxConnections parameter in the Configuration Registry.

Table 12.5-5. Check Database Connections - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Identify the values for database parameters associated with the appropriate Resource Planning application	**contact Database Administrator**
2	UNIX window (Planning/Management Workstation)	**single-click** or use procedure in Section 12.2.1
3	**isql –U <user ID> -S <database server>**	**enter text, press Enter**
4	**<database password>**	**enter text, press Enter**
5	**sp_who**	**enter text, press Enter**
6	**go**	**enter text, press Enter**
7	**sp_configure "user connections"**	**enter text, press Enter**
8	**go**	**enter text, press Enter**
9	**quit**	**enter text, press Enter**
10	Compare the number of actual connections with the number of connections for which the database has been configured	**read text**
11	Notify the Database Administrator of the results	**contact Database Administrator**